# Sinclair ZX Spectrum Userclub

Clubinfo Nr. 11(82) 01. November 1990

Rolf Knorre - Postfach 20 01 02 - 5600 Wuppertal 2
Telefon 0202 / 55 66 89
Stadtpsarkasse Wuppertal (BLZ 33050000) Konto Nr. 853382

Hallo Freunde,

das Titelbildproblem dürfte jetzt gelöst sein. Mein Entwurf, der mit dem letzten Info erstmalig aufgetaucht ist, fand allgemeine Zustimmung. Bis auf weiteres werde ich also dieses Bild verwenden.

Da es sonst mal wieder keine Neuigkeiten gibt, darf ich mich für heute verabschieden. Viel Spaß mit dem neuen Info.

Rolf Knorre

# Spectrum User Club Wuppertal

Wilhelm Wiecke, Lönsweg 92, 4904 Enger Tel. 05224/3449

Folge 15

Enger, den 24.10.90

Hallo Spectrum-User!

Auch diesen Monat läuft mir die Zeit wieder einmal davon, aber trotzdem will ich versuchen, meinen Beitrag noch rechtzeitig fertig zu bekommen. Natürlich geht's darin wieder um BETA-Disk! Kürzlich habe ich nämlich damit begonnen, mir ein Programm zu schreiben, daß die Inhalte der Organisations- und Katalog-Sektoren einer Disk vollständig in Klartext auf den Bildschirm oder Drucker bringt. Damit hat man dann eine noch vollständigere Übersicht über alle Filedaten, als es der LIST-Befehl der BETA-Versionen 5.xx bietet. Natürlich kann man das Programm auch in den Disk-Monitor integrieren oder es zu einem 'Regenerations-programm' für gelöschte Files ausbauen.

Da die Ausgabe natürlich schnell gehen soll, bietet sich dafür die Verwendung von Maschinensprache für den Hauptteil an, während das Steuerprogramm dazu in BASIC geschrieben wird. Was dabei bisher zustande gekommen ist, will ich im Folgenden vorstellen und da alles für einen Beitrag zu lang ist, geht's hoffentlich im nächsten Monat mit dem Rest weiter!

Also fange ich an mit den Lade- und Ausgabe-Routinen für den Org.-Sektor. Wie dieser aussieht und welche Daten er enthält, habe ich im Info 8/89 schon mal ausführlich erläutert. Deshalb gleich zum Assemblerlisting, daß hoffentlich jedem, der sich schon etwas mit Maschinensprache befaßt hat, halbwegs verständlich ist. Dazu habe ich es wieder mit reichlich Kommentaren versehen! Trotzdem einige zusätzliche Bemerkungen:

Als Startadresse habe ich hier 60000 gewählt (Zeile 100), als Pufferanfang für die Kopie der 9 Sektoren (8 Katalog- und 1 Org.-Sektor) Adr. 62000 (puf), sodaß dazwischen 2000 Bytes Platz für das MC-Programm bleiben. Das wird hoffenlich reichen. Es ist aber kein Problem, es an eine andere Laufadresse zu verlagern, wenn man ein Assemblerprogramm benutzt oder einen sogenannten Maschinencode-Verschieber! Anfangswerte müssen noch für die Laufwerksnummer (drive) und den Ausgabekanal (kanal) voreingestellt werden, die sich später auch aus dem BASIC-Steuerprogramm leicht ändern lassen. In Zeile 150 beginnt die eigentliche Programmabarbeitung mit einen Sprung (CALL sekload) in die Sektor-Laderoutine. Da ich auch diese schon im Info 1/90 beschrieben habe, kann ich mir das hier hoffentlich sparen. Nach Beendigung geht's zum Befehl 'JP orgaus', der nun die Ausgabe der Texte und Daten des Org.-Sektors auslöst.

Dazu wird zuerst der Bildschirm gelöscht, der Ausgabekanal (2 = Screen, 3 = Drucker) festgelegt und dann mit Hilfe einer ROM-Routine ($203C) der Kopftext ausgegeben. Da damit die Zeile beendet ist und noch eine Leerzeile folgen soll, wird sie 2* mit dem Code $0D = ENTER abgeschlossen (Zeile 1250/1260). Dann folgt die Ausgabe des Textes 'Diskname:', allerdings mit dem Unterschied, daß noch kein ENTER-Code die Zeile abschließt. Es muß schließlich erst noch der Name der Disk dahinter gesetzt werden. Damit die Ausgabe von Namen und Daten formatiert erfolgt, habe ich die Textzeilen jeweils mit Leerzeichen auf 23 Zeichen aufgefüllt, somit braucht die Zeilenlänge auch nicht für jeden Text einzeln neu ins BC-Reg. gebracht werden. Zuerst hatte ich eine andere Lösung vorgesehen und zwar sollte die PRINT-Pos. durch einen TAB-Befehl hinter dem jeweiligen Text festgelegt werden (also $17,$17 = TAB 23). Das funktionierte am Screen auch wunderbar, nur leider spielte der Drucker dann nicht so mit wie gewünscht. Der Ausdruck des Namens z.B. erfolgte statt an der angegebenen TAB-Position in einer neuen Zeile! Ich vermute, daß die eigentliche Ursache für die Störung in der Software des KEMPSTON E-Interfaces liegt, aber genaues weis ich noch nicht! (Der gleiche Effekt tritt auf, wenn der BETA-Disk-Befehl 'LIST #3' aufgerufen wird.)

Der auszugebende Diskname muß nun natürlich aus unserer Org.-Sektorkopie geholt werden. Dazu muß erst einmal die Anfangsadr. des ersten Zeichens ermittelt werden. Das ist jedoch eine einfache Rechnung, da dies immer das 245. Byte im Org.-Sektor ist. Weil vor diesem Sektor in der Kopie noch die die 8 Katalogsektoren stehen, müssen diese noch dazuaddiert werden. Somit ist das erste Zeichen des Disknamens das 2273. Byte im Puffer (8 * 256 +245 = 2293)! Dieser Wert wird in Zeile

600 zur Pufferanfangsadresse addiert und somit ist die benötigte Adresse ermittelt. Der Rest läuft wie schon erwähnt über die ROM-Routine 'Stringausgabe' ($203C) ab. Als Abschluß wird noch der ENTER-Code $0D angefügt (Zeile 630).

Bei den weiteren Texten läuft natürlich alles nach dem gleichen Schema ab, während die Zahlenausgabe zwangsläufig eine eigene Behandlung erfordert. Dazu muß wie beim Disknamen die Position des jeweiligen Wertes errechnet werden, allerdings wird nun statt der berechneten Adresse die Zahl auf die diese Adresse zeigt, ins HL-Reg. geladen. Die eigentliche Umwandlung der Hex.- in eine Dezimal-Zahl erfolgt mit Hilfe einer abgewandelten Routine aus meiner BETA-Disk-Version 5.03 (ab Zeile 1430). Eine ausführliche Erläuterung dazu habe ich bereits im Info 5/90 geliefert. Allerdings waren zwei kleine Änderungen notwendig.

Die Erste ist die Zeile 1430 (beim Label: 'zahl_1'), in der das H-Reg. gelöscht wird. Das ist hier immer erforderlich, wenn eine auszugebende Zahl <256 ist. Durch den Befehl 'LD HL,(puf+xxxx)' wird ja eine bei der errechneten Adresse beginnende 2-Bytezahl ins HL-Reg. gebracht, gebraucht wird aber davon nur das Low-Byte (z.B. beim ersten freien Sektor / Track, File-Gesamtzahl usw.)! Anders sieht es dagegen bei der Anzahl freier Sektoren aus. Da deren Zahl max. 2544 ($09F0) betragen kann, sind dafür 2 Bytes im Org.-Sektor reserviert, die natürlich auch ins dezimale Format gebracht werden müssen. Dazu erfolgt dann der Einsprung ins Umwandlungsprogramm in Zeile 1450 (Label: 'zahl_2')!

Eine weitere Änderung gegenüber der Originalroutine ist die Zeichenausgabe ab Zeile 2090 (Label: 'codeaus'). Wird dagegen Zeile 2050 angesprungen (Label: 'zeilend'), so wird der Code $0D als Zeilenabschluß ausgegeben, was ja in unserem Fall nach Ausgabe der letzten Stelle einer Zahl auch geschehen soll. Wie ich gerade sehe, hätte man auch die Befehle 'CALL codeaus' direkt durch 'RST $10' ersetzen können und damit je Befehl 2 Bytes Speicherplatz sparen können. Aber nun bleibt's erst mal so! Mancher wird sich schon gefragt haben, warum ich zur Zahlenausgabe überhaupt diese Routine verwendet habe wo doch die größte Zahl bisher nur 4-stellig sein kann. Dafür gibt's ja auch eine Routine im SPECTRUM-ROM ($1A1B, Zahl in BC übergeben) und die unterdrückt sogar führende Nullen! Der Grund ist der, daß im nächsten Teil, der Katalogausgabe, häufig 5-stellige Zahlen auftauchen (max. $FFFF = 65535) und da geht's dann nicht mehr ohne! Wenn mir noch eine elegante Lösung einfällt, liefere ich auch noch eine Ergänzung nach, mit der auch hier führende Nullen unterdrückt werden. Aber vielleicht hat ja auch jemand eine Idee dazu, und stellt seine Lösung mal vor. Ich würde mich sehr darüber freuen!

Eine Anmerkung noch zum Diskformat, dessen Codes bekanntlich $16, $17, $18 oder $19 lauten können. Deshalb habe ich auch in der Abfrage (Zeile 820-980) nur diese berücksichtigt. Eigentlich sollten keine anderen Codes bei BETA-Disk vorkommen und für Nicht-BETA-Disks ist das Programm sowie so nicht geeignet! Wer will kann aber auch hinter Zeile 960 mit 'CP $16' und 'JR Z,format' auf 80 Trk DS prüfen lassen, und für den Fall daß auch dieser Code noch nicht der richtige ist, eine Fehlermeldung ausgeben lassen und die Abarbeitung abbrechen. Aber eigentlich ist das nur sinnvoll, wenn man auf den Aufruf der Funktion 24 (=Change disk) in Zeile 310-320 verzichtet, denn dabei wird auch schon das Diskformat geprüft. Stimmt es nicht, so kommt es normalerweise schon dort zum Programmabbruch (näheres dazu siehe Info 11/89).

Ähnliches gilt auch für den Wert 'Sektoren pro Track'! Da steht bei BETA-Disk grundsätzlich $10 = 16, da keine andere Sektorenzahl vom TRDOS akzeptiert wird!

Im nächsten Teil kommt dann die ausführliche Auflistung der Katalogdaten dran. Dabei habe ich vor, die Angaben nach Filetypen getrennt exakt aufzuschlüsseln und damit die Daten im Katalog optimal nutzbar zu machen. Wenn nichts dazwischen kommt, wird hoffentlich alles rechtzeitig zu nächsten Info fertig, inclusive einem kleinen BASIC-Steuerprogramm. Wer an dem kompletten Programm interessiert ist, muß sich also noch ein wenig gedulden. Wenn es fertig ist, kopiere ich es gern kostenlos auf eine eingeschickte 3.5" Diskette (80 Trk SS oder DS, Format bitte angeben!) oder Kassette. Rückporto bitte nicht vergessen! Der abgedruckte Quelltext ist übrigens mit Hilfe des Assemblers 'MICASS' erstellt worden.

```
                0010 ;################################################################
                0020 ;#                                                              #
                0030 ;#        Funktion:  ORGANISATIONSSEKTOR-INHALTE AUFLISTEN      #
                0040 ;#        Filename: 'Orgaus-t'            Datum: 23.10.90       #
                0050 ;#                                                              #
                0060 ;#     Wilhelm Wiecke, Lönsweg 92, 4904 Enger, Tel. 05224/3449  #
                0070 ;#                                                              #
                0080 ;################################################################
                0090 ;
EA60            0100        ORG  60000      ;Startadresse beliebig!
                0110 ;
F230            0120 puf    EQU  62000      ;Pufferanfangsadresse
                0130 ;
                0140 ;
EA60 CD6FEA     0150        CALL sekload    ;Katalog- und Org.-Sektor laden!
EA63 C38EEA     0160        JP   orgaus     ;Danach Inhalt ausgeben!
                0170 ;
EA66            0180 drive  DB   0          ;Laufwerksnummer (kann ungepoket werden!)
EA67            0190 kanal  DB   2          ;Ausgabekanal-Nr.       '
                0200 ;
                0210 ;
                0220 ;UNTERPROGRAMM 'TEXTAUSGABE'
                0230 ;
EA68 011700     0240 textaus LD  BC,23      ;Textlänge laden
EA6B CD3C20     0250        CALL $203C      ;und ROM-Routine 'Stringausgabe' aufrufen zur Textausgabe!
EA6E C9         0260        RET             ;Zurück!
                0270 ;
                0280 ;KATALOG- UND ORG.-SEKTOR-INHALTE IN GESCHÜTZTEN SPEICHERBEREICH LADEN
                0290 ;
EA6F CD213D     0300 sekload CALL $3D21     ;TRDOS-Puffer vorhanden? Nein? Dann installieren!
EA72 0E18       0310        LD   C,24       ;Nr. der Funktion 24 (= Change disk) laden
EA74 CD133D     0320        CALL $3D13      ;und diese ausführen!
                0330 ;
EA77 3A66EA     0340        LD   A,(drive)  ;Laufwerksnummer ins A-Register (A=0, B=1, C=2, B=3) und
EA7A 0E01       0350        LD   C,1        ;C-Reg. mit Nummer der Funktion 'Laufwerksnummer wählen' laden!
EA7C CD133D     0360        CALL $3D13      ;Nun diese Funktion aufrufen und ausführen!
                0370 ;
EA7F 2130F2     0380        LD   HL,puf     ;Adr., an die der Sektorinhalt geladen wird.
EA82 0609       0390        LD   B,9        ;Zahl zu ladender Sektoren!
EA84 1600       0400        LD   D,0        ;Tracknr. des 1. zu ladenden Sektors in B- und Sektor-Nr. in
EA86 1E00       0410        LD   E,0        ;E-Reg., also hier 9 Sektoren (Trk 0 / Sek 0 bis Sek 8 laden).
EA88 0E05       0420        LD   C,5        ;Nun Funktionsnr. 5 (= Sektor lesen) in C-Reg. laden
EA8A CD133D     0430        CALL $3D13      ;und auch diese Funktion ausführen!
EA8D C9         0440        RET             ;Danach zurück!
                0450 ;
                0460 ;CLS, KANAL öFFNEN UND TITELTEXT-AUSGABE
                0470 ;
EA8E CD6B0D     0480 orgaus  CALL $0D6B     ;ROM-Routine CLS aufrufen!
EA91 3A67EA     0490        LD   A,(kanal)  ;Ausgabekanal festlegen (2 = Screen, 3 = Drucker).
EA94 CD0116     0500        CALL $1601      ;und gewählten Kanal öffnen!
                0510 ;
EA97 1125EB     0520        LD   DE,text1   ;Titeltext-Anfangsadresse in DE-
EA9A 012000     0530        LD   BC,32      ;und Textlänge in BC-Reg. bringen.
EA9D CD3C20     0540        CALL $203C      ;Nun ROM-Routine 'Stringausgabe' aufrufen!
                0550 ;
                0560 ;DISKNAMEN AUSGEBEN
                0570 ;
EAA0 1145EB     0580        LD   DE,text2   ;Textanfangsadr. von 'Diskname:' laden
EAA3 CD68EA     0590        CALL textaus    ;und zur Textausgabe!
EAA6 1125FB     0600        LD   DE,puf+2293 ;DE auf Disknamen-Anfang im 8. geladenen Sektor
EAA9 010800     0610        LD   BC,8       ;setzen und BC mit max. Zeichenanzahl für Namen laden!
EAAC CD3C20     0620        CALL $203C      ;Nun ROM-Routine 'Stringausgabe' aufrufen
EAAF CD6BEC     0630        CALL zeilend    ;und Zeile beenden!
                0640 ;
                0650 ;TEXTE UND ORG.-SEKTORDATEN-AUSGABE
                0660 ;
EAB2 115CEB     0670        LD   DE,text3   ;Textanfangsadr. von 'Erster freier Sektor' laden
EAB5 CD68EA     0680        CALL textaus    ;und zur Textausgabe!
EAB8 2A11FB     0690        LD   HL,(puf+2273) ;Nun Wert des 1. freien Sektors holen
EABB CD21EC     0700        CALL zahl_1     ;Sektornummer dezimal ausgeben!
                0710 ;
EABE 1173EB     0720        LD   DE,text4   ;Textanfangsadr. von 'Erster freier Track:' laden
EAC1 CD68EA     0730        CALL textaus    ;und zur Textausgabe!
EAC4 2A12FB     0740        LD   HL,(puf+2274) ;Nr. des 1. freien Trk. aus Org.-Sek.-Kopie holen,
EAC7 CD21EC     0750        CALL zahl_1     ;und Tracknummer dezimal ausgeben!
                0760 ;
EACA 118AEB     0770        LD   DE,text5   ;Textanfangsadr. von 'Diskformat:' laden
EACD CD68EA     0780        CALL textaus    ;und zur Textausgabe!
                0790 ;
                0800 ;NUN WIRD DAS DISKFORMAT ERMITTELT
                0810 ;
EAD0 3A13FB     0820        LD   A,(puf+2275) ;Code für Diskformat holen (mögliche Werte: $16 - $19.)
EAD3 11BCEB     0830        LD   DE,text6+27 ;Anfangsadr. von Text '40 Trk SS' laden
EAD6 010900     0840        LD   BC,9       ;Textlänge ist hier jeweils 9 Zeichen!
EAD9 FE19       0850        CP   $19        ;Ist der Formatcode im A-Reg. = $19?
EADB 2811       0860        JR   Z,format   ;Ja? Dann den Text ausgeben!
                0870 ;
EADD 11B3EB     0880        LD   DE,text6+18 ;Sonst Anfangsadr. von Text '80 Trk SS' laden
EAE0 FE18       0890        CP   $18        ;und prüfen, ob es Formatcode $18 ist.
EAE2 280A       0900        JR   Z,format   ;Ja? Dann den Text ausgeben!
```

```
EAE4 11AAEB  0910         LD   DE,text6+9   ;Sonst Anfangsadr. von Text '40 Trk DS' laden.
             0920 ;
EAE7 FE17    0930         CP   $17          ;Ist dies der richtige Formatcode?
EAE9 2803    0940         JR   Z,format     ;Dann diesen Text ausgeben! Sonst bleibt nur noch Code 16.
             0950 ;
EAEB 11A1EB  0960         LD   DE,text6     ;Deshalb Adr. von Text '80 Trk DS' laden.
EAEE CD3C20  0970 format   CALL $203C        ;Den ermittelten Text per ROM-Routine ausgeben
EAF1 CB6DEC  0980         CALL zeilend      ;und Zeile beenden!
             0990 ;
             1000 ;RESTLICHE TEXTE UND WERTE AUSGEBEN
             1010 ;
EAF4 11C5EB  1020         LD   DE,text7     ;Anfangsadr. von 'Files incl. gelöschte' laden
EAF7 CD68EA  1030         CALL textaus      ;und zur Textausgabe!
EAFA 2A14FB  1040         LD   HL,(puf+2276) ;Wert dazu ins HL-Reg. holen,
EAFD CD21EC  1050         CALL zahl_1       ;und den Wert als Dezimalzahl ausgeben!
             1060 ;
EB00 11DCEB  1070         LD   DE,text8     ;Anfangsadr. von 'Freie Sektoren' laden
EB03 CD68EA  1080         CALL textaus      ;und zur Textausgabe!
EB06 2A15FB  1090         LD   HL,(puf+2277) ;Zugehörigen Wert holen
EB09 CD23EC  1100         CALL zahl_2       ;und die Anzahl (hier eine 2-Byte-Zahl) dezimal ausgeben.
             1110 ;
EB0C 11F3EB  1120         LD   DE,text9     ;Anfangsadr. von Text 'Sektoren pro Track' laden
EB0F CD68EA  1130         CALL textaus      ;und zur Textausgabe!
EB12 2A17FB  1140         LD   HL,(puf+2279) ;Wert dazu holen (ist bei BETA-Disk immer 16)
EB15 CD21EC  1150         CALL zahl_1       ;und als Dezimalzahl ausgeben!
             1160 ;
EB18 110AEC  1170         LD   DE,text10    ;Textanfangsadr. von 'Gelöschte Files' laden
EB1B CD68EA  1180         CALL textaus      ;und zur Textausgabe!
EB1E 2A24FB  1190         LD   HL,(puf+2292) ;Auch hier den entsprechenden Wert holen,
EB21 CD21EC  1200         CALL zahl_1       ;und wiederum als Dezimalzahl ausgeben!
EB24 C9      1210         RET               ;Programm beenden!
             1220 ;
             1230 ;BEGINN DER TEXTTABELLE
             1240 ;
EB25         1250 text1    DM   '   ORG.-SEKTOR (TRK 0 / SEK 8)'
EB43         1260          DB   $0D,$0D      ;2 * <ENTER>
EB45         1270 text2    DM   'Diskname:           '
EB5C         1280 text3    DM   'Erster freier Sektor:  '
EB73         1290 text4    DM   'Erster freier Track:   '
EB8A         1300 text5    DM   'Diskformat:            '
EBA1         1310 text6    DM   '80 Trk DS40 Trk DS80 Trk SS40 Trk SS'
EBC5         1320 text7    DM   'Files incl. gelöschte: '
EBDC         1330 text8    DM   'Freie Sektoren:       '
EBF3         1340 text9    DM   'Sektoren pro Track:   '
EC0A         1350 text10   DM   'Gelöschte Files:      '
             1360 ;
             1370 ;AUSGABE EINER BIS ZU 5-STELLIGEN DEZIMAL-ZAHL
             1380 ;
             1390 ;Der Hex-Wert der umzuwandelnden Zahl muß hier im HL-Reg. stehen!
             1400 ;
             1410 ;Zuerst wird die ZEHNTAUSENDER-Stelle ermittelt!
             1420 ;
EC21 2600    1430 zahl_1   LD   H,0          ;Bei Zahl <256 das überflüssige High-Byte löschen!
             1440 ;
EC23 AF      1450 zahl_2   XOR  A            ;A-Reg. (und damit auch das C-Flag löschen)
EC24 111027  1460          LD   DE,$2710     ;und DE-Reg. mit 10000 laden!
EC27 EB52    1470 sub_zt   SBC  HL,DE        ;Vom momentanen Wert in HL nun 10000 abziehen
             1480 ;
             1490 ;Das Ergebnis der Subtraktion findet man anschließend im HL-Reg.!
             1500 ;
EC29 3803    1510          JR   C,zehntau    ;Wenn C-Flag gesetzt ist, Zehntausender-Stelle ausgeben.
EC2B 3C      1520          INC  A            ;Sonst A-Reg.-Inhalt um 1 erhöhen!
EC2C 18F9    1530          JR   sub_zt       ;Wiederholen, bis Wert in HL <0  ist (also quasi negativ!
             1540 ;
             1550 ;Dann wird das C-Flag (übertrags-Flag) gesetzt.
             1560 ;
EC2E C630    1570 zehntau  ADD  $30          ;A-Reg.-Inhalt + $30 = Code der 10000er-Stelle!
EC30 CD6FEC  1580          CALL codeaus      ;Nun zur Ausgabe des gefundenen Codes springen.
EC33 19      1590          ADD  HL,DE        ;Zum 'negativen' Wert in HL wieder 10000 addieren.
             1600 ;
             1610 ;Der verbliebene positive Rest befindet sich somit wieder im HL-Reg.!
             1620 ;
             1630 ;Nun die TAUSENDER-Stelle ermitteln!
             1640 ;
EC34 AF      1650          XOR  A            ;A-Reg. (und damit auch gleichzeitig das C-Flag löschen)
EC35 11E803  1660          LD   DE,$03E8     ;und DE-Reg. mit 1000 laden!
EC38 ED52    1670 sub_tau  SBC  HL,DE        ;Vom momentanen Wert in HL nun 1000 abziehen
EC3A 3803    1680          JR   C,tausend    ;Wenn C-Flag gesetzt ist, Tausender-Stelle ausgeben.
EC3C 3C      1690          INC  A            ;Sonst A-Reg.-Inhalt um 1 erhöhen!
EC3D 18F9    1700          JR   sub_tau      ;Wiederholen, bis Wert in HL <0 ist (also quasi negativ)
EC3F C630    1710 tausend  ADD  $30          ;A-Reg.-Inhalt + $30 = Code der 1000er-Stelle!
EC41 CD6FEC  1720          CALL codeaus      ;Nun zur Ausgabe des gefundenen Codes springen.
EC44 19      1730          ADD  HL,DE        ;Zum 'negativen' Wert in HL wieder 1000 addieren.
             1740 ;
             1750 ;Nun die HUNDERTER-Stelle ermitteln!
             1760 ;
EC45 AF      1770          XOR  A            ;A-Reg. (und damit auch gleichzeitig das C-Flag löschen)
EC46 116400  1780          LD   DE,$0064     ;und DE-Reg. mit 100 laden!
EC49 ED52    1790 sub_hun  SBC  HL,DE        ;Vom momentanen Wert in HL nun 100 abziehen
EC4B 3803    1800          JR   C,hundert    ;Wenn C-Flag gesetzt ist, Hunderter-Stelle ausgeben.
```

```
EC4D 3C        1810          INC  A           ;Sonst A-Reg.-Inhalt um 1 erhöhen!
EC4E 18F9      1820          JR   sub_hun     ;Wiederholen, bis Wert in HL <0  ist (also quasi negativ)
EC50 C630      1830 hundert  ADD  $30         ;A-Reg.-Inhalt + $30 = Code der 100er-Stelle!
EC52 CD6FEC    1840          CALL codeaus     ;Nun zur Ausgabe des gefundenen Codes springen.
EC55 19        1850          ADD  HL,DE       ;Zum 'negativen' Wert in HL wieder 100 addieren.
               1860 ;
               1870 ;Nun die ZEHNER-Stelle ermitteln!
               1880 ;
EC56 AF        1890          XOR  A           ;A-Reg. (und damit auch gleichzeitig das C-Flag löschen)
EC57 110A00    1900          LD   DE,$000A    ;und DE-Reg. mit 10 laden!
EC5A ED52      1910 sub_zeh  SBC  HL,DE       ;Vom momentanen Wert in HL nun 10 abziehen
EC5C 3803      1920          JR   C,zehn      ;Wenn C-Flag gesetzt ist, Zehner-Stelle ausgeben.
EC5E 3C        1930          INC  A           ;Sonst A-Reg.-Inhalt um 1 erhöhen!
EC5F 18F9      1940          JR   sub_zeh     ;Wiederholen, bis Wert in HL <0  ist (also quasi negativ)
EC61 C630      1950 zehn     ADD  $30         ;A-Reg.-Inhalt + $30 = Code der 10er-Stelle!
EC63 CD6FEC    1960          CALL codeaus     ;Nun zur Ausgabe des gefundenen Codes springen.
EC66 19        1970          ADD  HL,DE       ;Zum 'negativen' Wert in HL wieder 10 addieren.
               1980 ;
               1990 ;Nun bleibt als Rest noch die EINER-Stelle!
               2000 ;
EC67 7D        2010          LD   A,L         ;Rest (= 1er-Stelle) ins A-Reg. bringen!
EC68 C630      2020          ADD  $30         ;A-Reg.-Inhalt + $30 = Code der 1er-Stelle!
EC6A CD6FEC    2030          CALL codeaus     ;Nun zur Ausgabe des gefundenen Codes springen.
               2040 ;
EC6D 3E0D      2050 zeilend  LD   A,$0D       ;Code für )ENTER( laden,
               2060 ;
               2070 ;UNTERPROG. ZUR AUSGABE EINES ZEICHENS!
               2080 ;
EC6F D7        2090 codeaus  RST  $10         ;ROM-Routine zur Ausgabe des Codes im A-Reg. aufrufen!
EC70 C9        2100          RET              ;Zurück ins Hauptprogramm!

#      61A8  puf    F230  drive  EA66  kanal  EA67  textaus EA68  sekload EA6F  orgaus EA8E  format EAEE  text1  EB25
text2  EB45  text3  EB5C  text4  EB73  text5  EB8A  text6  EBA1  text7  EBC5  text8  EBDC  text9  EBF3  text10 EC0A
zahl_1 EC21  zahl_2 EC23  sub_zt EC27  zehntau EC2E  sub_tau EC38  tausend EC3F  sub_hun EC49  hundert EC50  sub_zeh EC5A
zehn   EC61  zeilend EC6D  codeaus EC6F
```

Und so sieht der Ausdruck eines Org.-Sektors im Klartext aus:

ORG.-SEKTOR (TRK 0 / SEK 8)

| | |
|---|---|
| Diskname: | Disk 36 |
| Erster freier Sektor: | 00014 |
| Erster freier Track: | 00131 |
| Diskformat: | 80 Trk DS |
| Files incl. gelöschte: | 00103 |
| Freie Sektoren: | 00450 |
| Sektoren pro Track: | 00016 |
| Gelöschte Files: | 00007 |

## SUCHMELDUNG ! ! !

**Wie schon im letzten Monat erwähnt, suche ich ein weiteres BETA-Disk- (möglichst Version 5.03, eventuell auch 4.xx) und ein Kempston-E Centronics-Interface. Außerdem interessiere ich mich für Schrittmotoransteuerung mittels SPECTRUM. Ich meine, auch dazu wäre mal ein Interface mit Motoren und Software im Handel gewesen. Wer so was hat und verkaufen möchte, kann mich ja mal anrufen (möglichst nach 18 Uhr)!**

So, damit soll's für diesen Monat wieder reichen. Bis demnächst!

U. Ul

Manfred Messerschmidt, Wildtaubenweg 64, 7050 Waiblingen
Tel. 07151/22387

Datenmanager 87 / kleine Veränderungen mit nützlichen Effekten 3. Teil

Als nächstes wenden wir uns dem Programmteil sequentielles suchen zu. Unter dem Buchstaben "F" ist es auf Diskette angelegt. Wie schon in den vorherigen Berichten beschrieben ist die Vorgehensweise die selbe.

Name: F

Neue Progr.-Länge: 1598 Nach dem Delete: 1568
Orginal-Länge: 1317

Erweiterung um: 261 Bytes

Mit der Programmlänge liegen wir hier noch weit unter der Teilprogrammlänge von Index-Suchen (G). Solange wir das längste Teilprogr. nicht überschreiten ist dies weiter nicht tragisch.

```
299 LET A=0: LET i=a: DELETE 299
  TO 299
```

```
/ Nach dem Aktivieren ist diese Zeile aus dem Progr. entfern
```

```
300 PROC cl: LET t=SNG PI: LET m
 askfl=a: PRINT AT a,a;" SUCHEN
  ": LET inv=a: LET c$="00016125
 5044": ROC kas: LET y$ =Unterbr
 echung der Suchoption durch drü
 cken der Taste 'q'": LET iy=21:
  LET ix=A: PROC d1
```

```
/
/ INFO-Zeile eingelegt
/ Bildschirm Position der INFO-Zeile durch eine Variable
/ hinterlegt. LET-Anweisung 'GLOBAL' entfernt sie liegt
/ ohne einen Nutzen im Programm.
```

```
305 LET i=CODE ">": LET Y$="0238
 0010063(G)LOBALSUCHE oder (F)EL
 DSUCHE :": PROC G: PROC INPUT:
 PROC K: IF NOT INSTRING(SNG PI,
 "FG",x$) THEN GO TO VAL "300":
 ELSE IF X$="G" THEN LET anf=SGN
  PI: LET end=bpr-SGN PI: LET CT
 YP=SGN PI: LET LA=32: GO TO VAL
  "315"
```

```
/
/ Variable 'i' bekommt die erste Zuweisung (Zeilenlänge)
/
/ Auch hier ist die unnötige LET-Anweisung 'GLOBAL' aus dem
/ Programm genommen.
/
```

```
315 LET j$="Suchbegriff :": IF l
 a>i THEN LET la=i: LET i=t
317 LET i=t: LET Y$="0415": IF l
 a>47 THEN LET y$="0501": LET i=
 a: PRINT AT 4,3;j$: ELSE LET i=
 t
318 IF ctyp THEN LET Y$=Y$+FN Y$
 (LA)+"0162"+(j$ AND i=t): LET i
 =t: PROC input: IF x$=STRING$(l
 a," ") THEN DIM k$(la): ELSE DI
 M k$(poi): LET k$=x$
```

```
/          Der komplette Block ist neu gestaltet
/
/ Variable 'i' muß zuerst zum Vergleichen herhalten. Ist das
/
/ Eingabefeld länger als 47 Zeichen, dann wird es eine Zeile
/
/ tiefer gesetzt. Der ungeschickte Zeilenumbruch des Ein-
/
/ gabefeldes entfällt.
```

```
355 LET iy=22: LET ix=CODE "4":
 PROC cl
```

```
/ Bildschirmposition für die Zählleiste
```

```
392 LET  Y$=USING$("0000.",i)+"
 Blatt": PROC d1: LET i=i+t
```

```
/ Zählleiste sie gibt Auskunft, wieviel Datenblätter mit dem
/ jeweiligem Suchkriterium über den Bilschirm gingen. Die
/ Variable 'i' dient nun als Zähler
```

```
393 PROC outrec
394 EXIT IF EDF
395 LOOP
396 PROC RESET: GO TO VAL "300"
```

```
/ Wegen einfügen der Zeile 392 ist dieser Block unnumeriert
/ und leicht platzsparend verändert.
```

Vor dem absaven Zeile 0 aus dem Programmteil nehmen.

---

Jetzt widmen wir uns dem der Eingabe. Alle Eingabedaten gehen über dieses Menueteil. Auf Diskette finden wir es mit dem Buchstaben 'A'.

Name: A

Neue Progr.-Länge: 2006 Nach dem Delete: 1972
Orginal-Länge: 1764

Erweiterung um: 208 Bytes

Die Erweiterung spielt auch hier keine große Rolle siehe Erklärung oben.

Hier alle zusammengefaßte Zeilen: 405+415; 465+470

```
299 LET a=SGN PI: LET t=NOT  PI:
  LET i=a: DELETE 299 TO 299
```

```
/ Nach dem Aktivieren ist diese Zeile aus dem Progr. entfernt.
```

```
300 CLS : PROC datnamdis: PRINT
 AT t,t;"Sätze eingeben": PROC
 RE: LET maskfl=t: LET g$=f$: LE
 T dasa1=dasa: DIM o$(dasa,bpr):
 LET ft=a: LET C$="000105160012"
 : PROC KAS: LET Y$="09010030063
 Anzahl der einzugebenden Sätze"
```

```
/
/ Die LET-Werte zwischen 't' und 'a' sind ausgetauscht. Jetzt
/
/ ist eine bessere Struktur vorhanden. Einige ehemalige LET-
/ Anweisungen liegen in der Zeile 299.
```

```
305 PROC input: IF VAL x$<VAL "1
 " THEN PROC w: GO TO VAL "305":
 ELSE LET nr=VAL x$
```

```
/ Zeile platzsparend verändert.
```

```
310 LET ip=t: LET inbc=DPEEK (Y(
 a)): PROC mes3
```

```
/ Zeile platzsparend verändert.
```

```
335 LET r$="": LET outbc=inbc-a:
 LET q=a: PROC c1
```

```
/ Zeile platzsparend verändert.
```

```
345 PROC inrec: LET R$=A$: PROC
 schreiben: DPOKE VAL "65482",DP
 EEK(VAL "65482")+a: PROC datnam
 dis
```

```
/ Zeile platzsparend verändert.
/
```

```
355 LET q=q+a
```

```
/ Zeile platzsparend verändert.
```

```
361 ON ERROR 2800: PROC schliess
 en: PROC reset: DIM o$(SGN PI):
 GO TO 300
```

```
/ Ohne ON ERROR steigt das Programm bei schreibgeschützte
/ Disketten aus.
```

```
392 IF INSTRING(a,"iu",n$(v,a))
 THEN LET A$(anf TO end)=MEMORY$
 ()(y(7) TO (y(8): LET ix=VAL SH
 IFT$(v)(3 TO 4): LET iy=VAL SHI
 FT$(v)( TO 2): LET Y$=A$(anf TO
  end): PROC d1: GO TO 402
```

```
/
/  Zeile platzsparend verändert.
/
/
```

```
395 BRIGHT a: PROC input: BRIGHT
  t: IF n$(v,a)="n" THEN ELSE GO
  TO VAL "400"

396 IF INSTRING(a,"aciopqstvzACI
 OPQSTVZ",x$(a)) THEN LET x$="r"

397 PROC ntest: IF NOT ko THEN P
 ROC W: LET i=t: PROC inf: PROC
 DATNAMDIS: GO TO VAL "395"

398 IF NOT i THEN PROC status: L
 ET i=a
```

```
/ Dieser Block ist verändert auch neue Teile sind eingefügt.
/ Das Feld in welchem gerade die Eingabe erfolgt ist aufge-
/ hellt dargestellt. Dadurch ist die Position des Cursor so-
/ fort sichtbar. Je nach Länge des Datenfeldes bzw. Lage, ka
/ es vorkommen, daß die Aufhellung um eine Zeichenlänge größ
/ ist. Eine Kontrolle für numerische Felder ist im Programm
/ enthalten. Jedoch sie funktionierte leider nicht richtig.
/ Alle Zeichen, die durch ein LET-Befehl einen Wert aufweise
/ werden nicht ausgesondert. Der eingelegte Wert wurde in da
/ Eingabefeld voll übernommen. Durch eine neu eingebrachte
/ Kontrolle kann dies nun ausgeschlossen werden. Bei Falsch-
/ eingabe erscheint eine Fehlermeldung.
```

```
402 PROG u: NEXT v
```

```
/  Schalterfunktion für das aufgehellte Feld.
```

```
430 IF NOT PEEK 55060 THEN IF ma
 skfl=3 THEN LET maskfl=t: ELSE
 LET maskfl=a
```

```
/ Zeile platzsparend verändert.
```

```
460 DEF PROC status: PROC c: LET
  h$="Eingabe : "+FN u$("000",q)
 "  F1 =Copy, F2 =Satz Nr.:": PR
 OC Prt: END PROC
```

```
/ Zeilenlöschfunktion eingefügt
/
```

```
480 IF h$="N" THEN PROC C: PROC
 Z4: LET j$="22000140063(W)eiter
 <> ENTER = Änderungs-": PROC ho
 lschluessel: PROC z2: PROC c: P
 ROC AENDERN: PROC test
```

```
/  Info-Zeile wegen vergrößerter Maske um eine Reihe tiefer
/  gesetzt.
```

```
495 DEF PROG aendern
```

```
/ Zeilenlöschbefehl entfällt.
```

```
505 LET y$=SHIFT$(o)+"0063": BRI
 GHT a: PROC input: BRIGHT t: PR
 OC u: IF n$(o,a)="n" THEN ELSE
 VAL "510"
506 IF INSTRING(a,"aciopqstvzACI
 OPQSTVZ",x$(a)) THEN LET x$="r"
507 PROC ntest: IF NOT ko THEN P
 ROC inf: PROC W: GO TO VAL "505
 "
```

```
/       Auch dieser Block ist neu gestaltet
/ Bevor das nachfolgende Datenblatt zum Eingeben aktiv wird
/ erscheint eine Sicherheitsabfrage. Hier können Fehler be-
/ hoben. Zur besseren Orientierung ist auch das Eingabefeld
/ aufgehellt.
/ Weitere Angaben findet man in den Zeilen  395-398.
```

```
650 DEF PROC u: ALTER PAPER VAL
 "7", BRIGHT a TO PAPER VAL "7",
 BRIGHT t: END PROC
```

```
/ Neue PROC-Routine sie dient zum Abdunkeln der aufgehellten
/ Eingabefelder.
```

```
670 DEF PROC inf: LET h$="Eingab
 e ist nicht nummerisch": PROC p
 rt: END PROC
```

```
/ Neue INFO-Zeile eingefügt.
```

Auch hier verfahren wir zum Saven wie oben beschrieben.

**Wer** arbeitet noch mit dem Daten Manager? **Suche** Erfahrungsaustausch.

**Zu verschenken:** 1 vollständig kopiertes Buch 'Weitere Kniffe und Programme mit dem ZX Spectrum' ca 110 Seiten. Portokosten müssen übernommen werden.
**Wer kann mir weiter helfen?** Für das Discovery soll es ein verbessertes Eprom geben. Durch eine geschicktere Programmierung sollen die Lade- bzw. Saveroutinen schneller von statten gehen.

---

Ich verkaufe als Paket (und nur als Paket):
1 Spectrum 48K Gummi, 1 Spectrum 48 K+, 1 defekten 48K Gummi, 1 Discovery 720K, 1 Ambermonitor mit Ständer, 1 ZX Printer, 1 DK Tronic Sprachmodul, 1 DK Tronic Syntheziser, 1 DK Tronic Light Pen, 1 Video Digitizer, Multiface one, diverse Joysticks, 1 Kassettenrecorder, Druckerkabel Software auf Diskette bis zum Abwinken, darunter Spiele aller Art und Anwendungen: TWIII, ART STUDIO, MASTERFILE, DTP und und und.... Zeitschriften: ASM (fast alle Ausgaben), Computer Kontakt, Mein Homecomputer, diverse engl. Zeitschr., Clubinfo's u.s.w. Bücher: Spezielle Bücher über den Specci

*Und das alles für lausige 600,-DM + Versandgebühr.*

Ich wünsche allen noch viel Spaß mit dem treuen Specci.

So long

**Uwe Gohmann  Dorfwiese 19  D-5600 Wuppertal 22  Tel.0202/602952**

# Text-o-Mat (10)

**Briefautomat:**
Wem ist es nicht leid, schnell mal irgendeinen Brief (Reklamation, Versicherung, Ämter usw.) zu schreiben, wenn man nicht täglich mit Schreibarbeiten zu tun hat. Es dauert länger sich dazu zu entschließen, als dann den Brief zu schreiben.
Eine weitere unangenehme Sache ist,immer wieder dasselbe schreiben zu müssen, immer wieder dieselben Fragen zu beantworten, immer wieder die gleichen Beschreibungen und Erklärungen abgeben zu müssen.
Da helfen die hier beschriebenen Routinen so sehr, daß man sich viel schneller entschließt, einen Brief zu schreiben, daß kaum noch Post unerledigt liegenbleibt. Selbst unangenehme Briefe kann man sich durch vorgefertigte Floskeln so erleichtern, daß sie die Erwartungsangst herabsetzen. Plötzlich gibt es keine Briefmuffel mehr, Der Club erhält eine Flut von Briefen und die Mitglieder vergessen nicht, sich zu bedanken, wenn sie von einem anderen Mitglied einen Hinweis oder gar ein Programm bekommen haben.
Nach dem Laden des Text-o-Mat geben wir Brightmode- a ein (Zeile 8234). Wir werden gefragt: An? und geben ein: h für Herrn, f:für Frau, m:für Fräulein. Vorname: Name: Straße: Ort mit Postleitz. Geben wir für Ort: m ein, dann werden wir nach dem Ortsteil gefragt, z.B.80. Der Computer notiert dann: '8000 München 80'. Geben wir: b ein, dann notiert der Computer: '1000 Berlin 80'. Wer mehr Platz hat, kann mehr Städte einbauen (ab Zeile 7062). Wir werden weiter gefragt nach Tit.:(Anrede) wenn: "j", dann schreibt der Computer an die richtige Stelle:
'Sehr geehrter Herr(Frau/Fräulein) Name,
Haben Sie vielen Dank für Ihre freundliche Anfrage vom'

Wird die Frage:Kopf? mit:"j" beantwortet dann wird ein Briefkopf mit Namen a zugeladen. Wird mit:"h" geantwortet, dann wird ein anderer Briefkopf von der Helpseite kopiert.
Die nächste Frage nach Datum, wird nur einmal beim ersten Brief gestellt, dann merkt sich der Computer das Datum auf Helpseite 2 unten, bis er wieder ausgeschaltet wird oder abstürzt.
Nach der Frage SAVE speichert er es aber soweit ab, daß nur jeweils der Tag eingegeben werden muß. So können wir Monat und Jahr ausgeschrieben eingeben, z.B. Sept. 1990 ohne es jedesmal schreiben zu müssen.
Am Ende dieser Fragen steht nun alles an der richtigen Stelle des Briefes und der Cursor ist dort, wo wir anfangen zu schreiben.
Der Briefkopf a ist geeignet für aufwendigere Briefköpfe mit Angaben über Konten o.ä. Einfach im Editor normal schreiben und mit Steuerzeichen versehen, dann abspeichern. Auch der Kopf: h kann mit Steuerzeichen versehen sein, soll aber wegen Platz und Aufwand einfacher sein.
Die eingegeben Adresse ist auf Helpseite 1, Zeile 1 gespeichert und kann durch Brightmode- k nochmal im Adressfenster des Briefes ausgedruckt oder mittels EXT-r- < in eine Liste gebracht werden.
Wenn wir nun öfter wiederkehrende Sätze oder Textteile haben, die wir immer wieder in unseren Briefen verwenden wollen, so können wir diese als Block abspeichern (Brightmode- B, S. 8180) und mit einem Namen aus 2 Buchstaben versehen. Wir können diese dann jederzeit mittels EXT- r- (zwei Buchstaben) an die Stelle des Cursors im Brief setzen (siehe Text-o-Mat (2),INFO 3/90 u.6/7/90).
Am Ende des Briefes geben wir EXT- r- g- SPACE oder EXT- r- g- h

ein und erhalten: 'mit freundlichen Grüßen
Name a oder Name h'
Mit EXT- r- v können wir einen kurzen Text an die Cursorposition bringen, der auf der Helpseite gespeichert ist. Der Cursor befindet sich dann hinter diesem.

Mit Brightmode- t erhalten wird am Ende des Blattes:
'Geschäftszeit und Telefondienst usw.'
Wir können uns auch einen aufwendigen Briefkopf mit and.Zeichensätzen oder Grafik mittels des zweiten Editors (Text-o-Mat (9)), LETTA-HEAD, DLAN oder anderen machen, mittels Brightmode- X,Zeile 8144,einladen, drucken und gleich darauf den Brief drucken, natürlich auch mehrere, wenn wir in Zeile 8144 mit INPUT den Namen eingeben.
Wir müssen also nicht, wie bei LETTA- HEAD, erst die Briefköpfe drucken und dann ein zweites Mal erst den Brief. Dann ist auch immer alles an der richtigen Stelle!
Adressen können auch aus Listen mit EXT- r-> auf die Helpseite gebracht und von dort mit Brightmode- k, Zeile 8078 an die richtige Position der Adresse gebracht werden. Nach Eingabe der ersten gewünschten Zeile, wählt der Computer bei Eingabe von w selbst die nächste Zeile, sodaß bis zu 26 Adressen selbsttätig eingegeben werden können.

**Herbert Hartig**
Postfach 147,8938 Buchloe

---

## Suche

dringend gebrauchten Specci !
Und träume von einer SAGA-Tastatur !

wer eines von beiden loswerden möchte :

Tel.: 0921/42653 (WG-Telefon)
Jürgen Schomburg
Dr.-Martin-Luther-Str.9
8580 Bayreuth

```
3015 IF INKEY$="v" THEN GO TO VAL "8189"
3020 IF INKEY$="g" THEN GO TO VAL "7120"
3022 IF INKEY$=";" THEN PAUSE NOT PI: LET i$=INKEY$: GO SUB VAL "7058": PRINT US
R VAL "55149",VAL "32e3"+z1,i$: GO TO VAL "3112"
7010 INPUT "Zeile?w";z$: IF z$="" THEN LET z$="-1"
7011 IF z$="w" THEN LET z$=STR$ (ze+zu)
7012 INPUT "Zahl?";y$;"adr. SAVE ";x$;: IF y$="" THEN LET y$="1"
7014 LET ze=VAL z$: LET zu=VAL y$: RETURN
7015 IF x$="s" THEN INPUT "Name:";n$: LET r=SQR PI: SAVE *r;n$CODE b,VAL "64"*zu
: GO TO VAL "3112"
7020 IF x$="a" THEN GO TO VAL "8079"
7022 RETURN
7050 IF e$="f" THEN LET e$="Frau"
7052 IF e$="h" OR e$="" THEN LET e$="Herrn"
7054 IF e$="m" THEN LET e$="Fräulein"
7056 IF h$="" THEN LET h$=" "
7057 IF f$="" THEN LET f$=" "
7058 IF i$="" THEN LET i$=" "
7060 IF i$="m" THEN GO SUB VAL "7098": LET i$="8000 München "+o$
7061 IF i$="b" THEN GO SUB VAL "7098": LET i$="1000 Berlin "+o$
7080 RETURN
  90 IF s$="s" THEN POKE VAL "57695",NOT PI: ERASE SGN PI;"tasword": SAVE *SGN P
I;"tasword"CODE VAL "54784",VAL "10751": VERIFY *SGN PI;"tasword"CODE
 92 RETURN
 95 IF d$="" THEN LET d$=" "
7096 RETURN
7098 INPUT "Ortst.?";o$: RETURN
7100 INPUT "T.0/Disk.1-2";r: RETURN
7110 IF k$="j" THEN LOAD *SGN PI;"a"CODE : RETURN
7112 IF k$="h" THEN GO SUB VAL "8049"
7114 RETURN
7120 PAUSE NOT PI
7122 PRINT USR VAL "55308".VAL "54959",z1+VAL "32e3",VAL "24"
7123 IF INKEY$="h" THEN PRINT USR VAL "55308",VAL "54867",z1+VAL "32320",VAL "14
": GO TO VAL "7127"
7125 PRINT USR VAL "55308",VAL "54984",z1+VAL "32256",VAL "32": PRINT USR VAL "5
5308",VAL "55016",z1+VAL "32384",VAL "14"
7127 LET a=USR VAL "64333": GO TO EXP PI
7918 PRINT USR VAL "55149",VAL "33152",q$: GO TO VAL "8008"
8078 GO SUB VAL "7010"
8079 PRINT USR VAL "55308",VAL "56129"+VAL "64"*ze,VAL "32640",VAL "23": PRINT U
SR VAL "55308",VAL "56152"+VAL "64"*ze,VAL "32704",VAL "20": PRINT USR VAL "5530
8",VAL "56172"+VAL "64"*ze,VAL "32832";VAL "20": GO TO PI
8144 INPUT "Kopien? ";z: GO SUB VAL "8145": GO TO VAL "8147"
8145 CLS : LOAD *"m";SGN PI;"Kopf5"SCREEN$ : RETURN : REM hier Name
  6 PRINT USR VAL "55555",VAL "3",NOT PI,VAL "6": LPRINT CHR$ VAL "27";"j";CHR$
VAL "189";: RETURN : REM Drucker zurück
  7 GO SUB VAL "8146"
48 POKE VAL "62235",SGN PI: LET a$="1": LET c=VAL "64"*(INT VAL a$-VAL "1"): L
ET b=c+FN p(VAL "62216"): LET x=VAL "60045": GO SUB VAL "950": LET b=a-c: IF z T
HEN GO SUB VAL "265"
8149 IF NOT z THEN GO SUB VAL "259": RUN
8150 LET z=z-SGN PI: IF z THEN GO TO VAL "8147"
8151 GO TO PI
8168 PRINT USR VAL "55308",VAL "55239",VAL "35840",VAL "63": GO TO PI
8180 INPUT "erste.";ER;"letzte Z.";LE;"Name? ";N$
8181 LET U7=VAL "31936"+(ER*VAL "64"): LET U8=VAL "31936"+((LE+1)*VAL "64"): LET
 U9=U8-U7: SAVE *SGN PI;N$CODE U7,U9: GO TO PI
8189 PRINT USR VAL "55308",VAL "54984",VAL "32e3"+z1,VAL "32": POKE VAL "62209",
PEEK VAL "62209"+VAL "33": LET a=USR VAL "64333": GO TO EXP PI
8234 INPUT "An:";e$;"Vorn.";f$;"Name";g$;"Str.";h$;"Ort";i$;"Tit.?(j)";j$;"Kopf?
";k$: GO SUB VAL "7110"
8235 IF NOT PEEK VAL "57695" THEN INPUT "Datum:";d$;" SAVE ";s$: GO SUB VAL "709
5": PRINT USR VAL "55149",VAL "57695",d$: GO SUB VAL "7090"
8236 GO SUB VAL "7050": PRINT USR VAL "55149",VAL "56193",f$: PRINT USR VAL "551
49",VAL "56203",g$: PRINT USR VAL "55149",VAL "56216",h$: PRINT USR VAL "55149",
VAL "56236",i$: PRINT USR VAL "55308",VAL "57695",VAL "33069",VAL "15": PRINT US
R VAL "55149",VAL "32576";e$+"     ": PRINT USR VAL "55149",VAL "32640",f$+g$: PR
INT USR VAL "55149",VAL "32704",h$: PRINT USR VAL "55149",VAL "32832",i$: LET q$
="Sehr geehrte": IF e$="Fräulein" AND j$="j" THEN LET q$="Liebes "+e$+" "+g$+",
": GO TO VAL "7918"
8237 IF e$="Herrn" THEN LET q$=q$+"r"
8238 IF j$="j" THEN LET q$=q$+" "+e$( TO VAL "4")+" "+g$+",": GO TO VAL "7918"
8239 GO TO PI
```

Hallo Rolf,

ich finde das Titelbild für das Info, so wie es im Oktober Heft dargestellt wird sehr gut. Du könntest meiner Meinung nach auch ruhig das Logo vom September Heft für jedes Info benutzen. Das sieht allemal besser aus als die Pubertären Strichmännchenzeichnungen ,die bisher das Info verunstaltet haben.

---

Nun suche ich noch Hilfe:

Ich habe seit ein Paar Wochen das PLUS D - Interface. Aber gerade die Software die von mir am meisten benutzt wird (TASWORD 128 und BETA BASIC 4.0) hat mit dem PLUS D die größten Schwierigkeiten. Bei TW128 ist es leider nicht möglich Textfiles auf Disc zu SAVEn bzw. zu LOADen, da diese Routinen vom Mcode aus auf das Interface 1 und die Microdrives zugreifen.

Es bleibt also nur die Cassette zum sichern und laden übrig. Dazu meine Frage: Hat einer der Clubfreunde vielleicht das Programm TW128 - Mcode mässig an das PLUS D angepasst? Ist über diese Problematik eventuell in der Zeitschrift OUTLET zu lesen (Herr Pirsch??). Wenn ja - für Tips und Pokes - oder auch diverse Maschinencode Routinen wäre ich sehr dankbar.

Auf den Artikel von Guido Schell habe ich mir bei Uffenkamp das Programm TASWORD III bestellt. Die Software kam tatsächlich mit einer 3 1/2 Diskette an. Nach dem Laden musste ich aber feststellen, daß das Programm für meine Konfiguration untauglich ist. Wegen meines seriellen Druckers ist mir bei TW128 noch möglich über die 128er Spectrum RS232 auszudrucken. Bei TW III - das nur im 48K-Mode läuft- kann ich aber diese RS232 nicht ansprechen. Jetzt würde also nur ein neuer Drucker helfen; jedoch - sagen wir 400 DM für einen Drucker - dazu die DM 88.41; die Tasword III gekostet hat. Das sind dann schon fast 500 Märker!!

---

Hallo Manfred Messerschmidt,
ist es eventuell möglich das Programm DM87 an die PLUS D - Syntax anzupassen? Bei PLUS D ist Load & Save mit LOAD d1"name" - oder SAVE d1"name" möglich. Die Ströme werden wie beim Interface 1 angesprochen.

Beispiel: OPEN #4;D1"filename" IN oder OUT je nachdem ob ein File geschrieben oder gelesen werden soll. Geschlossen wird ein Strom mit CLOSE #4.

Sollten mit der OPus-Version solche Ähnlichkeiten bestehen, so lass mich doch das wissen.

Für die Hilfe schon jetzt vielen Dank im voraus und Servus bis zum nächsten mal.

Karl-Heinz Meanger - Am Schloßberg 19 - 8431 Hohenfels - 09472/609

# Spectrum User Club Wuppertal

---

KARL-HEINZ MEANDER * AM SCHLOßBERG 19 * 8431 HOHENFELS

---

Hallo Spectrumfreunde,

für dieses Info habe ich mal wieder ein kleines BASIC-Strategiespiel als Beitrag gewählt. Das Programm ist für die heutige Zeit zwar nicht mehr überwältigend - aber was soll's; es ist halt wieder ein bischen Futter für den Spectrum.

Goldmine ist ein Strategiespiel, in dem man die Inhaberrolle einer Minengesellschaft übernimmt. Die Aufgabe ist, möglichst hohe Gewinne zu erwirtschaften.
Strategiespiele wie auch Adventures sind oft in BASIC geschrieben, weil die Geschwindigkeit der Assembler-Programmierung unnötig ist. Das Programm GOLDMINE wurde mit Grafik versehen, die den Abbau zeigt. Zu Beginn des Spiels besitzt man bereits die Bergwerksgesellschaft und zwei Millionen Dollar in bar. Dieses Geld soll in die Goldsuche investiert werden. Es sind dann in 30 Runden so viel Gewinn wie möglich zu erzielen. Spielen kann man allein oder gegen Jemand, der ein Konkurrenzunternehmen hat. (1 oder 2 Spieler).
Bevor man eine Mine ausbeuten kann, muss erst ein passendes Gelände gefunden werden. Dafür muß ein Gutachten angefordert werden, welches etwas über die Wahrscheinlichkeit einer erfolgreichen Grabung sagt. Es ist dann zu entscheiden, ob sich die Ausbeutung der Mine lohnt.
Hat man sich für ein Gelände entschieden, findet dort aber kein Gold, kann man an einer anderen Stelle erneut sein Glück versuchen.

Wurde Gold gefunden, so kann man das abgebaute Gold entweder im Tresor verwahren oder sofort an der Börse verkaufen. Bargeld hat Vorteile; allerdings gibt es Goldräuber, die den Schatz auf listige Weise rauben könnten.

---

Soweit als die kleine Spielebeschreibung.

Nun zum Infoheft 10/90: Ich besitze nun seit ein Paar Tagen das PLUS D - Interface. Ursprünglich wollte (und will noch immer) ich mir das DISCiPLE kaufen; jedoch war keiner der deutschen Soft & Hardware - Häuser bereit mir eines zu bestellen (Warum??). Im großen und ganzen muß ich dem Testbericht von Guido Schell zustimmen. Es fängt an jeden Tag mehr und mehr Spass zu machen wenn man mit dem PLUS D arbeitet.

Eines ist allerdings ärgerlich. Das PLUS D erlaubt kein ablegen der TASWORD 128 Files auf Diskette. Man ist also gezwungen die Textfiles auf Cassette zu saven! Dazu eine Frage an Guido: Ist dieses Tasword für PLUS D eine 128K-Version?

Das Beta Basic mit dem PLUS D nicht so richtig arbeiten will, ist verständlich. Da arbeiten sicher viele Routinen des BB Mcodes mit dem IF1-ROM zusammen. Es gibt aber eine an DISCiPLE und PLUS D angepasste Version von Beta Basic (Quelle Beta Basic Newsletter) und ich habe mir bei der Fa. Reitemann schon eine solche Version bestellt.

Zum Bericht von Angela: Es wäre nicht schlecht wenn Du im nächsten Info veröffentlichen würdest, wie man ein solches Abo bestellt bzw. wie eine solche Abo-Bestellung auszusehen hat. Ich glaube kaum, dass es reicht auf eine Postkarte nur einfach SERVUS zu schreiben. Also liebe Angela...

Dank an den unbekannten User für die POKEs zum Höher Formatieren!

```
  1>RESTORE 8000: FOR n=USR "a" TO USR "o"+7: READ a: POKE n,a: NEXT n
  5 BORDER 6: PAPER 6: INK 0: CLS
 10 PRINT AT 9,2;"How many Players?(1 or 2)": LET a$=INKEY$: IF a$="" THEN GO T
O 10
 20 IF a$<"1" OR a$>"2" THEN GO TO 10
 30 LET p=VAL a$: LET nop=p
 40 DIM a(2,6): DIM c(2,5): DIM a$(p,8): DIM r(2): LET er=10000
 50 LET r(1)=0: LET r(2)=0: LET a(1,1)=2000000: LET a(1,2)=2000000: LET a(2,1)=
2000000: LET a(2,2)=2000000: LET a(1,4)=100000: LET a(2,4)=100000:
 70 FOR n=1 TO p: INPUT "Name of Player ";(n);"?", LINE a$(n): NEXT n
200 FOR n=1 TO 30: FOR m=1 TO nop
202 BORDER 7: PAPER 7: INK 0: CLS
210 PRINT PAPER 6;n: PRINT PAPER 1; INK 6;AT 0,6;"  G O L D M I N E  "
220 PRINT TAB 16;a$(1);: IF nop=2 THEN PRINT TAB 24;a$(2);
230 PRINT '"TOTAL ASSETS $";TAB 15;a(1,1);: IF nop=2 THEN PRINT TAB 24;a(2,1);
240 PRINT '"CASH ASSETS  $";TAB 15;a(1,2);: IF nop=2 THEN PRINT TAB 24;a(2,2);
250 PRINT '"GOLD ASSETS kg";TAB 15;a(1,3);: IF nop=2 THEN PRINT TAB 24;a(2,3);
260 PRINT '"COST TO MINE $";TAB 15;a(1,4);: IF nop=2 THEN PRINT TAB 24;a(2,4);
270 PRINT '"NO. OF MINES";TAB 15;a(1,5);: IF nop=2 THEN PRINT TAB 24;a(2,5);
280 PRINT '"MINE DEPTH   m";TAB 15;a(1,6);: IF nop=2 THEN PRINT TAB 24;a(2,6)'
300 PRINT '' PAPER 4; INK 0;"Current Exchange Rate:-": PRINT "$";er;" per kg
gold"
400 PRINT ' PAPER 5;">-";a$(m)
500 PRINT PAPER 2; INK 7;"1";: PRINT "-Research and Development"
510 PRINT PAPER 2; INK 7;"2";: PRINT "-Exploration and Report"
520 PRINT PAPER 2; INK 7;"3";: PRINT "-Increase mine depth by 200m"
530 PRINT PAPER 2; INK 7;"4";: PRINT "-Exchange gold for dollars"
540 PRINT PAPER 2; INK 7;"5";: PRINT "-Pass"
550 PRINT : PRINT FLASH 1; PAPER 1; INK 6;"Enter your instruction"
600 LET i$=INKEY$: IF i$="" THEN GO TO 600
610 IF i$<"1" OR i$>"5" THEN GO TO 600
620 GO SUB VAL i$*1000
700 IF a(m,2)<0 THEN GO TO 7000
710 LET er=er+INT (RND*1000)-200
720 IF INT (RND*1600)-a(m,3)<0 THEN GO SUB 900
740 LET a(m,1)=a(m,2)+a(m,3)*er
750 PAPER 7: INK 0: BORDER 7: CLS
790 NEXT m
800 NEXT n
810 BORDER 5: PAPER 5: INK 0: CLS
820 PRINT FLASH 1; INK 7; PAPER 2;AT 6,10;" GAME OVER "
830 PRINT TAB 5;"Total Assets of ";a$(1): PRINT TAB 11;"$";a(1,1)
840 IF nop=2 THEN PRINT TAB 5;"Total Assets of ";a$(2): PRINT TAB 11;"$";a(2,1)
850 PRINT PAPER 2; INK 6; FLASH 1;TAB 2;"Press any key to Play again"
860 IF INKEY$<>"" THEN GO TO 860
870 IF INKEY$="" THEN GO TO 870
880 RUN
900 PAPER 2: INK 6: BORDER 2: CLS
905 LET jk=INT (RND*100)+50: IF jk>a(m,3) THEN LET jk=a(m,3)
910 PRINT PAPER 6; INK 1; FLASH 1;AT 9,8;" R O B B E R Y "
920 PRINT : PRINT INK 7;"      You have had ";jk;" kg of": PRINT "    your gold
assets stolen": LET a(m,3)=a(m,3)-jk: LET a(m,1)=a(m,1)-(jk*er)
930 FOR x=1 TO 35: BEEP .05,40: BEEP .05,20: NEXT x
940 BORDER 7: PAPER 7: INK 0: CLS : RETURN : REM mit IF1 genuegt CLS #
1000 BORDER 6: PAPER 6: INK 0: CLS
1010 PRINT PAPER 1; INK 6;AT 3,4;" RESEARCH & DEVELOPMENT ";AT 4,4;"(to lower mi
ning costs) "
1020 PRINT AT 7,6;"How much would you";TAB 5;"like to invest ? ($)": INPUT rd
1050 LET a(m,4)=a(m,4)-INT (rd*.05)-1
1060 IF a(m,4)<0 THEN LET a(m,4)=0
1080 LET a(m,2)=a(m,2)-rd: LET a(m,1)=a(m,1)-rd
1100 PRINT AT 13,3;"Your mining costs will be";TAB 3;"reduced by $";INT (rd*.05)
+1;" per 200m"
1110 PRINT '''"press a key": PAUSE NOT PI
1120 RETURN
2000 PAPER 4: BORDER 4: INK 0: CLS
2030 LET r(m)=0: LET c(m,1)=INT (RND*90)+10: LET c(m,2)=INT ((RND*5)+2)*200: LET
```

```
2030>LET r(m)=0: LET c(m,1)=INT (RND*90)+10: LET c(m,2)=INT ((RND*5)+2)*200: LET
 c(m,3)=INT (RND*200)+1: LET ii=INT (RND*3)-1
2050 LET c(m,4)=c(m,2)+ii*200
2070 LET c(m,5)=0: LET kk=INT (RND*100): IF kk<c(m,1) THEN LET c(m,5)=1
2080 PRINT PAPER 6; INK 0;AT 2,6;" SCIENTIFIC REPORT ": PRINT AT 5,5;"Chance of
Gold= ";c(m,1);" %": PRINT AT 7,5;"Expected Depth= ";c(m,2);" m": PRINT AT 9,5;"
Expected amount= ";c(m,3);" kg"
2100 LET z=INT (RND*150000): LET a(m,2)=a(m,2)-z: LET a(m,1)=a(m,1)-z
2110 PRINT FLASH 1;AT 12,0;"Would you like to mine? (y or n)"
2120 LET r$=INKEY$: IF r$="" THEN GO TO 2120
2130 IF r$="y" OR r$="Y" THEN LET a(m,6)=0: LET r(m)=1: GO TO 3000
3000 BORDER 6: PAPER 6: INK 0: CLS
3010 IF r(m)=0 THEN PRINT FLASH 1;AT 9,2;"You have not explored yet!": FOR z=1 T
O 10: BEEP .3,-10: NEXT z: RETURN
3020 BORDER 5: PAPER 4: CLS
3022 PRINT PAPER 5;TAB 14;CHR$ 147;CHR$ 148;CHR$ 149;TAB 14;CHR$ 150;CHR$ 151;CH
R$ 152;CHR$ 153;TAB 13;CHR$ 154;CHR$ 155;CHR$ 156;CHR$ 157;CHR$ 158;TAB 31;CHR$
 159
3025 FOR z=1 TO 32: PRINT CHR$ 144;: NEXT z
3050 PRINT AT 4,0;: FOR z=100 TO 1400 STEP 100: PRINT TAB 4-LEN STR$ z;z: NEXT z
3090 LET a(m,2)=a(m,2)-a(m,4): LET a(m,1)=a(m,1)-a(m,4): LET a(m,6)=a(m,6)+200:
PAUSE 30
3100 PRINT AT 3,15;CHR$ 146: FOR f=4 TO (a(m,6)/100)+3: PRINT AT f,15;CHR$ 145:
FOR w=1 TO 10: BEEP .01,-20: NEXT w: NEXT f
3120 IF a(m,6)=c(m,4) AND c(m,5)=1 THEN GO TO 3500
3130 PRINT FLASH 1; PAPER 5;AT 6,18;"No gold yet!": IF a(m,6)=c(m,2)+200 THEN PR
INT FLASH 1; PAPER 1; INK 6;AT 18,0;"This mine doesn't contain any  gold. Try s
tarting another one. ": FOR z=1 TO 10: BEEP .5,-20: NEXT z: LET a(m,6)=0: LET r(
m)=0
3140 PAUSE 150
3300 RETURN
3500 PRINT PAPER 6; INK 2; FLASH 1;AT f,12;"G O L D": FOR z=-20 TO 50: BEEP .017
,z: NEXT z: PAUSE 75
3550 LET a(m,5)=a(m,5)+1: LET a(m,3)=a(m,3)+c(m,3): LET a(m,1)=a(m,1)+(a(m,3)*er
): LET a(m,6)=0: LET r(m)=0: GO TO 3300
4000 PAPER 6: INK 1: BORDER 6: CLS
4020 PRINT INVERSE 1;AT 2,7;" EXCHANGE AGENCY ";AT 6,0;"The current exchange rat
e is:-";AT 8,5;"1 kg of gold= $";er;AT 12,2;"Enter no. of kg to exchange": INPUT
 nte
4040 IF nte>a(m,3) THEN PRINT FLASH 1;AT 16,0;"You do not have that much gold!"
4050 LET nte=INT nte
4090 IF nte>a(m,3) OR nte<0 THEN GO TO 4020
4095 PRINT AT 16,0;CHR$ 32;TAB 31;CHR$ 32
4100 LET a(m,3)=a(m,3)-nte: LET a(m,2)=a(m,2)+(nte*er): LET a(m,1)=a(m,1)+(nte*e
r)
4130 PRINT PAPER 5;AT 16,1;nte;" kg exchanged for $ ";nte*er: PAUSE 170: RETURN
5000 RETURN
7000 PAPER 5: INK 0: BORDER 5: CLS
7010 PRINT AT 9,12;a$(m): PRINT AT 10,8;"has gone bust!": PRINT FLASH 1;AT 20,1;
" Press any key to Play again "
7030 PAUSE NOT PI: RUN 5
8000 DATA 255,85,170,0,0,0,0,0,62,28,56,126,28,62,120,28
8010 DATA 255,255,62,126,127,60,124,126,0,0,0,0,1,1,1,1
8020 DATA 7,29,49,45,255,255,91,126,128,96,48,80,152,140,252,138
8030 DATA 1,1,1,49,49,49,49,255,122,187,62,95,153,255,153,126
8040 DATA 209,177,224,128,128,128,128,128,0,0,128,128,64,32,32,16
8050 DATA 1,3,7,7,4,4,7,7,255,255,255,255,149,149,159,159
8060 DATA 24,126,153,255,126,153,126,219,128,192,224,240,248,168,248,255
8070 DATA 16,8,8,4,14,31,31,255
```

## HALLO FREUNDE

Danke für die vielen Blumen. Ich habe sie an den Specci weitergegeben. In letzter Zeit gab es jede Menge Post und noch mehr Fragen. Da ich nicht nur Graphik mache (allerdings mein neuestes Hobby), möchte ich eine besonders interessante herausgreifen: Dtp, Md-Version, editor!

Richtig, Freunde, der Cursor im editor! strapaziert selbst den schnellsten Finger. Er ist noch schneller und nur deswegen klappt es auch mit dem Stick nicht. Immer wieder schieße ich über's Ziel. Frust (mein Mädchen behauptet, das sei eine Abkürzung: Finger runter, sonst tuscht's!) auf der ganzen Strecke. Hilfe muß her. Hier kommt sie schon in Form einer einfachen Verzögerungsschleife.

vorher

| | | | |
|---|---|---|---|
| D45A | FF | RST | #38 |
| D45B | FF | RST | #38 |
| D45C | FF | RST | #38 |
| D45D | FF | RST | #38 |
| D45E | FF | RST | #38 |
| D45F | FF | RST | #38 |
| D460 | FF | RST | #38 |
| D461 | FF | RST | #38 |
| D462 | FF | RST | #38 |
| D463 | FF | RST | #38 |
| D464 | FF | RST | #38 |
| D465 | FF | RST | #38 |
| D466 | FF | RST | #38 |
| D467 | FF | RST | #38 |
| D468 | FF | RST | #38 |
| D469 | FF | RST | #38 |

nachher

| | | | |
|---|---|---|---|
| D45A | F5 | PUSH | AF |
| D45B | C5 | PUSH | BC |
| D45C | E5 | PUSH | HL |
| D45D | D5 | PUSH | DE |
| D45E | 06FF | LD | B,#FF |
| D460 | 11FEFE | LD | DE,#FEFE |
| D463 | 21FF4B | LD | HL,#4BFF |
| D466 | 19 | ADD | HL,DE |
| D467 | 38FD | JR | C,#D466 |
| D469 | 10F8 | DJNZ | #D463 |
| D46B | D1 | POP | DE |
| D46C | E1 | POP | HL |
| D46D | C1 | POP | BC |
| D46E | F1 | POP | AF |
| D46F | 21AB63 | LD | HL,#63AB |
| D472 | C9 | RET | |

Natürlich sollten wir auch dem editor! sagen, wo die Bremse ist. Bitte schön:

vorher

| | | | |
|---|---|---|---|
| 638C | 21075C | LD | HL,#5C07 |
| 638F | 7E | LD | A,(HL) |
| 6390 | E6DF | AND | #DF |
| 6392 | 3600 | LD | (HL),#00 |
| 6394 | 21AB63 | LD | HL,#63AB |

nachher

| | | | |
|---|---|---|---|
| 638C | 21075C | LD | HL,#5C07 |
| 638F | 7E | LD | A,(HL) |
| 6390 | E6DF | AND | #DF |
| 6392 | 3600 | LD | (HL),#00 |
| 6394 | CD5AD4 | CALL | #D45A |

Tief gibt es dazu nicht zu sagen, zu einfach ist die Melodie. Kleinere Werte in d464 lassen den Cursor schneller blinken und genau so schnell "nervöser" wirken. Mit einem langsameren editor! werdet ihr Eure Fonts wesentlich angenehmer erstellen. Es ist schon paradox: Durch Verzögern kommt man besser voran (Beamtenregel).

Die Fragen zur Graphik lassen sich ganz kurz beantworten:
Decor-, Shade- und Corner-Sachs sind Dtp-Graphikfiles (mit einem Sternchen voran), Ink ist 0, Paper 7, damit Ihr den Schattierer (über den Headliner! erreichbar) nach Eurem Geschmack einsetzen könnt. Border-Sachs arbeiten im Textmodus, sind also "Zeichensätze". Die "Tastaturschablone" ist immer dabei, damit Ihr wißt, welches graphische Muster auf welcher Taste liegt. Im Info 6/7 war schon ein recht einfaches Beispiel. Demnächst mehr zum Thema Dtp.

Herzliche Grüße,

Walter Sperl

CLUBINFO - Nr. 11 / November 1990

Hallo Rolf, fur Deine Misere bezuglich der Titelbilder, sende ich Dir eine mogliche Titelbildgestaltung meines Sohnes daß vielleicht die ZX Spectrum User überraschen wird. Es geht wie man sieht auch mit dem Spectrum 48k.
Zugegeben es war eine mühselige Arbeit (Zeitaufwand ca. 6 Std. Aber es geht ohne (Amiga, Atari, Euro-Pz. usw.)
Meine Frage im Zusammenhang mit diesem Bild? Wer aus der Spec= trum - Gesellschaft hat ein Programm, daß den Drucker veranlaßt das konstruierte Bild in ähnlicher Größe, wie auf dem Monitor zu sehen ist auszudrucken.
Es wurde mich freuen, hier die eine oder andere Anregung zu er= halten.
Unser Drucker ist der SEIKOSHA GP-500AS.

übrigens, die Speichererweiterung, Druckerkabel, sowie das neue ROM (fur SAM) habe ich erhalten und bereits eingebaut. Der Beschaffer meiner Utensilien war einmal mehr KAI UFFENKAMP, dem aus meiner Sicht einzigen Handler (SPectrum-SAM) in naher Zukunft bei dem etwas zu bekommen ist.

Sieglinde Georg

# Spectrum User Club Wuppertal

Hubert Kracher
Kirchdorfer Str. 5
8201 Großholzhausen

Harald Lack
Heidenauer Str. 5
8201 Raubling

Hallo Specci-Freaks!!

## Der Befehlssatz des Zilog Z 80 / Teil 4

### DAA

Dezimalanpassung des Akkumulators.

Dieser Befehl addiert bedingt "6" zum rechten und/oder linken Nibble des Akkumulators, abhängig vom Inhalt des Statusregisters. Dieser Befehl dient zur BCD-Umwandlung nach arithmetischen Operationen.

Beispiel: DAA

### DEC m

Dekrementiere Operand m.

Der Inhalt der Speicherstelle, die durch den angegebenen Operanden adressiert wird, wird dekrementiert und wieder an dieser Stelle gespeichert.

Beispiel: DEC C

### DEC rr

Dekrementiere Registerpaar rr.

Der Inhalt des angegebenen Registerpaares wird dekrementiert und wieder in dem Registerpaar gespeichert.

Beispiel: DEC BC

### DEC IX

Dekrementiere IX.

Der Inhalt des Registers IX wird dekrementiert und das Ergebnis wieder im Register IX gespeichert.

Beispiel: DEC IX

### DEC IY

Dekrementiere IY.

Der Inhalt des Registers IY wird dekrementiert und das Ergebnis wieder im Register IY gespeichert.

Beispiel: DEC IY

### DI

Sperre Interrupt.

Die Interrupt Flip-Flops werden zurückgesetzt und damit alle maskierbaren Interrupts gesperrt. Nach Ausführung dieses Befehls ist der maskierbare Interrupt gesperrt bis er durch einen Befehl EI wieder freigegeben wird.

### DJNZ e

Dekremtiere B und springe falls nicht 0.

Das Register B wird dekrementiert. Ist das Ergebnis nicht 0, dann wird der Offset mit Zweierkomplement-Arithmetik zum Befehlszähler addiert, so daß sowohl vorwärts als auch rückwärts gesprungen werden kann. Der Offset wird addiert zu PC+2 (nach dem Sprung). Deshalb liegt der effektive Offset zwischen -126 und +129 Bytes. Der Assembler subtrahiert automatisch vom Quell-Offset, um den Hex-Code zu erzeugen.

Beispiel: DJNZ $ - 5 ($ = current PC)

### EI

Gib Interrupt frei.

Die Interrupt Flip-Flops werden gesetzt und dadurch nach der Ausführung des Befehls, der auf EI folgt, maskierbare Interrupts freigegeben. Bis zu diesem Zeitpunkt sind maskierbare Interrupts gesperrt.

Beispiel: Eine übliche Folge am Ende einer Interruptroutine:

```
EI
RETI
```

Nach Ausführung von RETI sind maskierbare Interrupts wieder freigegeben.

### EX AF,AF'

Vertausche Akkumulator und Flags mit den Zweitregistern.

Der Inhalt von Akkumulator und Statusregister wird mit dem Inhalt der entsprechenden Zweitregister vertauscht.

Beispiel: EX AF,AF'

### EX DE,HL

Vertausche Register DE und HL.

Der Inhalt von Registerpaar DE und Registerpaar HL wird vertauscht.

Beispiel: EX DE,HL

### EX (SP),HL

Vertausche HL mit dem Inhalt des obersten Stapelelements.

Der Inhalt des Registers L wird mit dem Inhalt der Speicherstelle vertauscht, die der Stapelzeiger adressiert. Der Inhalt des Registers H wird mit dem Inhalt der Speicherstelle



Spectrum User Club Wuppertal

vertauscht, die auf die durch den Stapelzeiger adressierte folgt.

Beispiel: EX (SP),HL

EX (SP),IX

Vertausche IX mit dem Inhalt des obersten Stapelelementes.

Der Inhalt der unteren Hälfte des Registers IX wird mit dem Inhalt der Speicherzelle vertauscht, die durch den Stapelzeiger adressiert wird. Der Inhalt der oberen Hälfte des Registers IX wird mit dem Inhalt der Speicherzelle vertauscht, die unmittelbar auf die durch den Stapelzeiger adressierte folgt.

Beispiel: EX (SP),IX

EX (SP),IY

Vertausche IY mit dem Inhalt des obersten Stapelelementes.

Der Inhalt der unteren Hälfte des Registers IY wird mit dem Inhalt der Speicherzelle vertauscht, die durch den Stapelzeiger adressiert wird. Der Inhalt der oberen Hälfte des Registers IY wird mit dem Inhalt der Speicherzelle vertauscht, die unmittelbar auf die durch den Stapelzeiger adressierte folgt.

Beispiel: EX (SP),IY

EXX

Vertausche mit den Zweitregistern.

Der Inhalt der Universalregister wird mit dem Inhalt der entsprechenden Zweitregister vertauscht.

Beispiel: EXX

HALT

Halte CPU an.

Die CPU (Mikroprozessor/Central Processing Unit) unterbricht die Operation und führt so lange NOP's (No Operation) aus, bis ein Interrupt oder ein Reset eintritt, um weiterhin Memory Refresh Zyklen zu liefern.

Das war's für heute. Bis zum nächsten Teil.



# Spectrum User Club Wuppertal

## Spectrum User Club Wuppertal

## Fragebogenaktion 1990

Hallo Leute !

Vor der eigentlichen Auswertung eine Adressliste von Specci-Händlern. Ich habe die Händler nicht alle überprüft, daß heißt ich weis nicht ob noch alle Specci-Sachen haben. Am besten selbst nachfragen. Bei Uffenkamp gibt es auf jeden Fall noch was. Unter anderem viele AMX-Mäuse und GP50 Drucker (Klopapierdrucker). Nun die Adressen:

Kai Uffenkamp, Gartenstr. 3, 4904 Enger

Auteq, Hofheimerstr. 53, 6238 Hofheim 7

T.S. Datensysteme, Denisstr. 45, 8500 Nürnberg 80
(Restbestände)

Computer Accessoires Int'l, Kreuzstr. 13, 8000 München 2
(Ersatzteile ???)

Worldwide Software, 106a Chilwell Road, Beeston,
Nottingham NG9 1ES, Great Britain

Gebrauchte Sachen findet man in der Zeitschrift
Computerflohmarkt (neues Heft Ende Oktober)

Sitronic GmbH, Volkersdorfer Str.1, 2732 Sittensen
(Cartridges ???)

E. Reitemann, Füllenbachstr.11, 4000 Düsseldorf 30

Wenn jemand Glück bei obigen Adressen hat, bitte unbedingt im Info mitteilen !!!

Viel Erfolg

Guido aus Lünne

Frank-Michael MOCZKO

Haselsteig 41
1000 Berlin 47

Berlin,06.10.90

## PASCAL ohne Strings ???

Der Variablentyp **string** erlaubt die Verarbeitung von Texten. Leider stellt HiSoft-Pascal diesen Typ nicht vordefiniert zur Verfügung.
Eine Stärke von Pascal ( und von unserem SPECTRUM ) ist es, daß wir diese Arbeit selbst machen können :

```
10   CONST N = 8  ; ( * DAS IST EIN BEISPIEL! *)
20   TYPE STRING = ARRAY [1..N] OF CHAR;
30   VAR NAME  : STRING ;
```

Die Definition in Zeile 20 enthält die vorgesehene Länge des strings ; ( um das Programm gut leserlich zu machen und auch pflegeleicht (!), wird die Länge in die Konstantendeklaration Zeile 10 ausgelagert ).

Will man eine Variable über die Tastatur einlesen , dann muß man ( ich verweise auf meinen Artikel in der User-Club - Zeitung #74, 2,1990 ) unbedingt die Befehlsfolge **readln; read (name)** verwenden, weil sonst die Zuordnung der Variablen nicht korrekt erfolgt.

**WICHTIG :**

Im Gegensatz zu anderen Pascal-Compilern ( z.B.: UCSD ) darf der String aber weder länger noch kürzer sein als die Definition !
Im ersten Fall wird das Zuviel ignoriert , im zweiten Fall werden die fehlenden Zeichen (char) bei der Ausgabe durch Fragezeichen '?' ergänzt, z.B.: PETER ergibt bei der Ausgabe ( gemäß der obigen Definition ) PETER??? ( also insgesamt 8 Zeichen).

Wie umgeht man nun die störenden '?' in der Ausgabe ?

Ein Beispielprogramm soll das erläutern :

```
10   PROGRAM A;
20   CONST N = 10;
30   TYPE STRING =ARRAY [ 1..N ] OF CHAR;
40   VAR NAME: STRING;
50       I   : INTEGER;
60   BEGIN
70   READLN;
80   READ(NAME);
90   FOR I:= 1 TO N DO
100       IF NAME[I] =CHR(0) THEN
110       NAME[I] := ' ';  (* ZWISCHEN DEN BEIDEN HOCHKOMMATA IST
                              EIN LEERZEICHEN BLANK *)
120  WRITELN(NAME:6);
130  END.
```

Mit diesem Programm kann man also einen bis zu 10-Zeichen langen string einlesen und dann ausgeben.

Für eine Liste der Länge 12 von Namen , die höchstens 8 Buchstaben lang sein dürfen und die z.B. in umgekehrter Reihenfolge wieder ausgegeben werden soll, kann das Listing so aussehen:

```
 10  PROGRAM B;
 20  CONST N = 8;
 30  TYPE   STRING = ARRAY [ 1..N ] OF CHAR;
 40  VAR    NAME : ARRAY [ 1..12 ] OF STRING;
 50         I,J:INTEGER;
 60  BEGIN
 70   FOR I:=1 TO 12 DO
 80       BEGIN (*FOR*)
 90       READLN;
100       READ(NAME[I]);
110       FOR J:=1 TO N DO
120       IF NAME[I][J] = CHR(0) THEN
130       NAME[I][J] :=' ' ;
140       END; (*FOR*)
150  WRITELN;
160  WRITELN('************************************');
170  FOR I:=12 DOWNTO 1 DO
180  WRITELN(NAME[I]:10);
190  END.
```

In den Zeilen 120/130 kann man statt Name[I][J] auch Name[I,J] verwenden!

Die in UCSD-Pascal für die **string**-Bearbeitung vorhandenen Funktionen **length**, **pos**(ition), **concat** (verknüpfen) , **copy** und **delete** muß man sich bei Bedarf also selbst herstellen :
Als Beispiel: **length** (ermittelt die Länge eines strings)

```
Kopf wie Programm b;
50         I,J,Z:INTEGER;
..
      Z:=0;
      FOR I:= 1 TO  N  DO
      IF NAME [I] = CHR(0) THEN Z:= Z+1 ;
      WRITELN(' DIE LAENGE DES NAMENS IST : ', N-Z : 5 );
..
```

Einer Variablen ( hier z.B.: NAME in Program b ) kann auch ein Wert zugewiesen werden :

```
NAME :=  'asdfghjk';     (* das sind genau 8 Zeichen *)
```

Stimmen Wortlänge und Definition nicht überein, erfolgt bei der Compilierung eine Unterbrechung und Fehlermeldung error 10.

Mit dem hier vorgestellten Werkzeug kann man nun beginnen, Namenslisten alphabetisch zu bearbeiten und auszugeben .

Also , bis dann
mit vielen Grüßen aus Berlin Michael

Thomas Hentschel,Hartwigstr.6,3000 Hannover1,0511/18639
Guten Tag dem Restclub,
trotz Einheit kein Wachstum im Club ( d( derswo, vielleicht finden ja die russischen Spectrumfreunde mal dazu. Egal, zeigt sich doch , daß auch 8-Bitter ihren Platz haben, besonders der Z80 findet immer neue Freunde. Die letzte Generation von Amateurfunktransceivern hat ihn eingebaut bekommen, russische und englische Spectrumclones tauchen auf (gesehen hab ich noch keinen, Erfahrungsberichte würden mich sehr interessieren) und auch bei Sir Sinclair tut sich was in dieser Richtung. In der letzten "Personal Computer World" las ich ein Interview, in dem er seine Zukunftspläne von sich gab: Neben Elektroauto und Klappfahrrad (grün) auch Computerhigh und -lowtech, Parallelrechner und Tragbares. Geplant ist eine Kreuzung aus Z88 und PSION-Portable. Alles noch etwas neblig, wohl MSDOS-kompatibel, aber auch mit integrierter Software und Stromspar-technik a la Z88. Auf letzerem schreib' ich gerade, der hat eine wunderbare Textverarbeitung im ROM, die eigentlich aber ein Tabellenkalkulationsprogramm ist. überhaupt eine unterschätzte Maschine und hier kaum verbreitet. In England scheint zumindest jeder EDV-Journalist einen zu haben. Kein umwerfernder kommerzieller Erfolg, aber lt.Sinclair auch in Zukunft "aktiv unterstützt", offenbar einer seiner Lieblinge. Zu recht, deshalb hier einige Bemerkungen. Zuerst: Hard und -Software sind ausgereift. Weder bei mir noch bei dem Rechner eines Bekannten seit 1 1/2 Jahren Probleme mit der Schaltung oder den Programmen, und die sind recht komplex. Die Tabellenkalkulation rechnet vordefinierte oder selbstprogrammierte Funktionen quer durch alle Zellen, die Formeln oder Text enthalten können. Deshalb läuft dieses Programm auch als Textverarbeitung (mit dem schönen Namen "Pipedream"). Als solche bietet sie alle features professioneller Software. Die Größe der Dokumente ist nur vom verfügbaren Speicherplatz abhängig, und der ist um 3 MB erweiterbar. Das Betriebssystem kann 4 MB adressieren und läuft auf einer CMOS Z80-CPU. Allerdings sah ich bislang nur maximal 1/2 MB Speichereinschübe, und auch die gehen ins Geld. Außerdem ist es nicht sehr sinnvoll, den 3.Speicherplatz mit RAM-Karten zu belegen, weil dort ein EPROM-Brenner eingebaut ist. EPROMs sind die einzige interne und leider teure Möglichkeit, Daten über einen längeren Stromausfall zu retten. Da bietet sich die eingebaute V24-Schnittstelle für eine Rechnerkopplung doch an. Für PCs, MacIntoshs, Acorns gibt's das zu kaufen, für den SPECTRUM zum Abtippen. Der elektrische Anschluß ist einfach, Z88 (links vom Pfeil) ist DTE und SPECTRUM-IP1 (rechts) bekanntlich DCE: 2->2, 3<-3, 4->4, 5<-5, 7-7. Mit der folgenden Software lassen sich dann Daten in beide Richtungen übertragen, editieren und auf Cassette oder MD abspeichern (nicht SPECTRUM-Programme, die Betriebssysteminformationen werden nicht erfaßt).

Die ESC-Codes entsprechen dem Z88 Import/Export-Protokoll. Import/Export ist eines der eingebauten Popdowns, das bei Kommunikation über die serielle Schnittstelle automatisch bestimmte Steuercodes generiert und auch über diese gesteuert werden kann im Gegensatz zur Terminal-Applikation, die angesagt ist, wenn die Gegenseite dieses Protokoll nicht unterstützt. Vor einer Datenübertragung müssen die Parameter der Schnitt- stelle eingestellt werden, der Z88 arbeitet mit maximal 38400 Baud, XON/XOFF und Parität sind drin. Eingestellt wird's im Panel-Popdown, wo auch die Tastatur-, Dateiverwaltungs-und Datenformatsparameter definiert werden. Andere Popdowns sind Uhr, Kalender, Wecker (kann auch CLI-Dateien ausführen), Taschenrechner (mit 10 Speicherplätzen und eiem Menü, das englische Maßeinheiten umrechnet), die Dateiverwaltung (mit hierarchischer Dateistruktur ähnlich MSDOS, wildcards, CLI...) und der Index, der umfangreiche Informationen über die Projekte des Benutzers bereitstellt. Im Gegensatz zu den Popdowns, die aus jeder anderen Aktivität heraus aufgerufen werden können und nur einmal vorhanden sind, ist es möglich, mehrere Dokumente einer Applikation parallel zu bearbeiten. So kann man zwischen verschiedenen natürlich mehrseitigen Texten hin und herspringen oder in der Textverarbeitung einen Programmtext editieren und schnell mal im Basic laufen lassen. Das eingebaute BBC-Basic hat kaum noch mit dem SPECTRUM-Basic zu tun. Es hat viele Befehle zur Dateibearbeitung, läßt eine PASCAL-ähnliche Programmstruktur zu (Prozeduren, globale und lokale Variable) und hat einen integrierten Z80-Assembler. Das Handbuch allerdings kann man in diesem Punkt völlig vergessen, ohne das spezielle BBCBASIC(Z80)-Handbuch (teuer) wird es frustig. Wenn ich da ans SPECTRUM-Handbuch denke... Zwie Applikationen kommen noch dazu, das programmierbare Tagebuch (noch nie benutzt) und der/die Druckertreiber. Von Haus aus an den EPSON-Standard angepaßt, lassen sich sehr einfach Neue programmieren und abspeichern. Zwischen die Programme und den Benutzer hat CAMBRIDGE COMPUTER LTD dankenswerterweise noch einen CLI gelegt. Damit ist es möglich, über Befehlsdateien Routinejobs zu automatisieren, zwischen den Programmen hin und herzuschalten, den Rechner fernzusteuern oder sich freundlich eine gute Nacht wünschen zu lassen. Der Phantasie sind sind kaum Grenzen gesetzt, und so soll es sein. Leider ist die Maschine hier recht teuer (900,-), wenn man mit der englischen Version zufrieden ist, kann man sie im UK für ca.600,- bestellen. Dort findet man auch Pakete mit Rechnerkopplungen, RAMs und Eproms und Netzteil, die preislich dann unserem Einzelverkaufspreis entsprechen. Cassettenrecorder und Diskdrives, die über die V24-Schnittstelle betrieben werden, gibt's auch. Aber der gute alte SPECTRUM macht sowas vielleicht besser, sicher billiger. Hier also das Programm, viel Spaß!

Tschüs Thomas


V851/...
V8256.....

excl. VAT

.....£174.00
mmer Pack .....£212.00
Pack PC .....£235.00
Pack Mac.....£275.00
l for accessories

HARD CARDS

21Mb Filecard.....£184.00
32Mb Filecard.....£214.00
40Mb Filecard.....£239.00

SAME DAY DELIVERY AVAILABLE IN LONDON AREA
081-777 7447
FAX NO: 081-777 3715
VMS Enterprises Ltd, The Warehouse
2a Braemar Gardens, West Wickham, Kent BR4 0JW.


(PCW 10/90)


MBRIDGE COMPUTERS
) SINCLAIR PRODUCTS

SWANLEY ELECTRONICS
Dept PCW, 32 Goldsel Rd., Swanley, Kent BR8 8EZ, England
SWANLEY (0322) 64851. We give an additional discount of 2% to personal callers
l orders welcome. UK prices are shown first and include post and VAT. The second
in brackets is for export customers in Eire and Europe and includes insured airmail
ce. The third price is for export customers outside Europe and includes insured airmail


(PCW 10/90)

# Spectrum User Club Wuppertal

```
   1 REM Thomas Hentschel:cancom,Hannover 1990
   2 REM Programm zur Rechnerkopplung von ZX Spectrum mit Interface1 und Z88
   9 REM Menu
  10 CLS : INPUT ; INVERSE 1;TAB 2;"Zeichen vom Z88 laden:";TAB 28;" <1>"''TAB 2
;"auf MD abspeichern:";TAB 28;" <2>"''TAB 2;"auf Cassette abspeichern:";TAB 28;"
 <3>"''TAB 2;"laden vom MD:";TAB 28;" <4>"''TAB 2;"laden von Cassette:";TAB 28;"
 <5>"''TAB 2;"senden zum Z88:";TAB 28;" <6>"''TAB 2;"SPECTRUM-Basic zum Z88:";TA
B 28;" <7>"''TAB 2;"Zeigen/editieren:";TAB 28;" <8>"; INVERSE 0''TAB 30;menu
  20 IF menu=1 THEN CLEAR : GO SUB 100: GO TO 10
  30 IF menu=2 THEN GO SUB 300: GO TO 10
  40 IF menu=3 THEN GO SUB 400: GO TO 10
  50 IF menu=4 THEN CLEAR : GO SUB 500: GO TO 10
  60 IF menu=5 THEN CLEAR : GO SUB 600: GO TO 10
  70 IF menu=6 THEN GO SUB 700: GO TO 10
  80 IF menu=7 THEN GO SUB 1500: GO TO 10
  85 IF menu=8 THEN GO SUB 1000: GO TO 10
  90 GO TO 10
  99 REM Zeichen vom Z88
 100 CLS : INPUT ;TAB 5;"Z88:"'' INVERSE 1;TAB 4;"AUF iler"''TAB 4;"CATALOGUE FI
LES"''TAB 4;"Name : <Laenge>"; INVERSE 0''''''''TAB 4; INVERSE 1;"wieviele Zeich
en?:"; INVERSE 0;TAB 23;dim
 110 CLS : PRINT ''''TAB 4;"Datei vom Z88 laden:"''''''''''''TAB 2;"weiter im Z8
8..."''''"Imp-Export aufrufen: "; INVERSE 1;" AUX "; INVERSE 0''TAB 21; INVERSE 1
;"S)end file "; INVERSE 0
 120 FORMAT "b";1200
 130 CLOSE #4
 140 OPEN #4;"b"
 150 LET dim=dim+50: DIM z$(dim)
 160 LET x=1
 170 LET a$=INKEY$#4
 180 IF CODE a$=0 THEN GO TO 170
 190 LET z$(x)=a$
 200 IF CODE z$(x)=69 THEN GO TO 260
 210 LET x=x+1
 220 LET zeich=x
 230 GO TO 170
 260 IF CODE z$(x-1)=27 THEN CLOSE #4: GO SUB 1000: RETURN
 270 GO TO 210
 299 REM auf MD abspeichern
 300 CLS
 310 INPUT ; INVERSE 1;TAB 3;"Laufwerk Nr.? :"; INVERSE 0;TAB 22;md: CAT md
 320 INPUT ; INVERSE 1;TAB 3;"Name der Datei? :"; INVERSE 0;TAB 22;f$
 330 SAVE *"m";md;f$ DATA z$()
 340 VERIFY *"m";md;f$ DATA z$()
 350 CLS : PRINT "OK"
 360 PAUSE 60
 390 RETURN
 399 REM auf Cassette abspeichern
 400 CLS
 410 INPUT ; INVERSE 1;TAB 3;"Name der Datei? :"; INVERSE 0;TAB 22;f$
 420 CLS : PRINT ;TAB 3; INVERSE 1;"EAR-Stecker rausziehen!..."; INVERSE 0
 430 SAVE f$ DATA z$()
 440 CLS : PRINT ;TAB 3; INVERSE 1;"...EAR-Stecker wieder rein."; INVERSE 0''TAB
 3; INVERSE 1;"PLAY druecken."; INVERSE 0
 450 VERIFY f$ DATA z$()
 460 CLS : PRINT "OK"
 470 PAUSE 60
 490 RETURN
```

**162**

**Clive on his bike**

Sir Clive Sinclair, everyone's favourite boffin, has dropped out of the headlines but he has some exciting projects planned; projects which, he claims, will change the world. Simon Rockman talks to the man who is planning a comeback.

(PCW 10/90)

```
 499 REM laden vom MD
 500 CLS
 510 INPUT ;TAB 3; INVERSE 1;"laden von MD Nr.? : "; INVERSE 0;TAB 26;md: CAT md
 520 INPUT ;TAB 3; INVERSE 1;"Name der Datei? :    "; INVERSE 0;TAB 25;f$
 530 LOAD *"m";md;f$ DATA z$()
 590 RETURN
 599 REM laden von Cassette
 610 INPUT ;TAB 3;"laden von Casette:"''TAB 3; INVERSE 1;"Name der Datei? : "; I
NVERSE 0;TAB 22;f$
 620 LOAD f$ DATA z$()
 630 RETURN
 699 REM senden zum Z88
 700 CLS
 710 INPUT ;TAB 3; INVERSE 1;"Name der Datei?:"; INVERSE 0;TAB 20;f$
 720 FORMAT "b";1200
 730 CLOSE #4
 740 OPEN #4;"b"
 750 PRINT #4;CHR$ 27;CHR$ 78;f$: REM ESC N ame
 760 PRINT #4;CHR$ 27;CHR$ 70: REM ESC F ile
 770 LET x=1
 780 PRINT #4;z$(x);
 790 LET x=x+1
 800 PRINT #4;z$(x);: IF z$(x)="E" AND CODE z$(x-1)=27 THEN GO TO 820
 810 GO TO 790
 820 PRINT #4;CHR$ 27;CHR$ 90: REM ESC Z uende
 830 CLOSE #4
 840 RETURN
 999 REM zeigen
1000 CLS
1010 POKE 23692,0
1020 LET x=1
1030 IF CODE z$(x)=13 THEN PRINT CHR$ 13;: GO TO 1070
1040 IF CODE z$(x)<32 OR CODE z$(x)>127 THEN PRINT ;" "; INVERSE 1;CODE z$(x); I
NVERSE 0;" ";: GO TO 1070
1050 PRINT z$(x);
1060 IF z$(x)="E" AND CODE z$(x-1)=27 THEN LET zeich=x: PAUSE 0: GO SUB 1200: RE
TURN
1070 LET x=x+1
1080 GO TO 1030
1199 REM editieren
1200 CLS
1210 INPUT ;TAB 3; INVERSE 1;"editieren? : <j>"; INVERSE 0;TAB 22;j$: IF j$<>"j"
 THEN RETURN
1220 FOR x=1 TO zeich
1230 PRINT ;TAB 3; INVERSE 1;x;":"; INVERSE 0;" ";
1240 IF CODE z$(x)<32 OR CODE z$(x)>127 THEN PRINT ; INVERSE 1;CODE z$(x); INVER
SE 0
1250 IF CODE z$(x)>=32 AND CODE z$(x)<=127 THEN PRINT z$(x)
1260 INPUT ;TAB 3; INVERSE 1;"letztes Zeichen aendern?: <j>"; INVERSE 0''TAB 29;
j$
1270 IF j$="j" THEN GO TO 1300
1280 NEXT x
1290 RETURN
1300 INPUT ;TAB 3; INVERSE 1;"neues Zeichen? : " ; INVERSE 0;TAB 22;w$
1310 LET z$(x)=w$
1320 GO TO 1280
1499 REM SPECTRUM-Basic zum Z88
1500 PRINT '''TAB 4;"SPECTRUM-Basic zum Z88:"
```

# Spectrum User Club Wuppertal

```
~~INT ''TAB 4;"SPECTRUM-Basic zum Z88:"~~
1510 INPUT ;TAB 3; INVERSE 1;"laden von Cassette:";TAB 23;" <c> "; INVERSE 0''TA
B 3; INVERSE 1;"laden vom MD:";TAB 23;" <m> "; INVERSE 0''TAB 24;1$
1520 IF 1$="c" THEN CLS : PRINT ;TAB 3; INVERSE 1;" PLAY druecken "; INVERSE 0:
LOAD "specbas"
1530 IF 1$="m" THEN INPUT ;TAB 3; INVERSE 1;"MD Nr.?: "; INVERSE 0;TAB 14;md: LO
AD *"m";md;"specbas"
1540 RETURN
```

```
9999 CLS : INPUT ;TAB 3;"SPECTRUM-Basic zum Z88:"''TAB 3; INVERSE 1;"MD Nr.?:";
INVERSE 0;TAB 12;md''TAB 3; INVERSE 1;"Name des Programms?:"; INVERSE 0''TAB 11;
p$: MERGE *"m";md;p$: LIST : FORMAT "t";1200: FORMAT "b";1200: CLOSE #3: CLOSE #
4: OPEN #3;"t": OPEN #4;"b": PRINT #4;CHR$ 27;CHR$ 78;p$;CHR$ 27;CHR$ 70: LLIST
: PRINT #4;CHR$ 27;CHR$ 90: CLOSE #3: CLOSE #4: REM specbas
```

```
9999 CLS : INPUT ;TAB 3;"SPECTRUM-Basic zum Z88:"''TAB 3; INVERSE 1;"Name des
ogramms?: "; INVERSE 0''TAB 3;p$: MERGE p$: LIST : FORMAT "t";1200: FORMAT "b";1
200: CLOSE #3: CLOSE #4: OPEN #3;"t": OPEN #4;"b": PRINT #4;CHR$ 27;CHR$ 78;p$;C
HR$ 27;CHR$ 70: LLIST : PRINT #4;CHR$ 27;CHR$ 90: CLOSE #3: CLOSE #4: REM specba
s
```

Noch ein paar Erläuterungen zum Programm:

Mit <BREAK> GOTO1 läßt es sich immer neu starten, ohne Daten zu verlieren.

Beim Senden vom SPECTRUM zum Z88 zuerst den Z88 anwerfen (Im/Export: B) atch receive), sonst bekommt er den Start irgendwie nicht mit...mal sehen.

Beim Senden vom Z88 zum SPECTRUM wird nach der Anzahl der Zeichen gefragt, die ist im FILER-Popdown zu erfahren. Normalerweise reicht dieser Wert, bricht cancom mit "Subscript wrong, 190:1" ab, war es doch nicht genug. Dann einfach die Prozedur mit einem höheren Wert nochmal fahren, zuviel schadet nicht. Grund dafür ist, daß der Z88 Steuerblöcke mitüberträgt, die je nach Struktur des Textes unterschiedlich lang sind. Außerdem werden bestimmte Bereiche des Zeichensatzes mit ESC B HEX-codiert übertragen. Das wird natürlich länger: aus "ö" wird "27 B F6". Natürlich kann auch das Programm an der entsprechenden Stelle geändert werden (einen höheren Wert addieren), ich mag nicht.

Die obere Version von "specbas" (Zeile 9999) gehört als separates Programm aufs MD, die untere ebenso auf Cassette. Die SPECTRUM-Programme sollten natürlich keine Zeile 9999 haben.


**CAMBRIDGE**

**Z88**
32k RAM. 8 Line, 80 Character supertwist LCD display. RS232 Port. Built-in software. Word processor, Spreadsheet, Database, Calendar/Diary, Calculator, Clock, Alarm. £175

**Z88 Summer Pack**
Features as Z88 but with 128k RAM Pack, Mains adaptor, Carry case, Eprom Eraser, 32K Eprom. £213

**Z88 Link Pack PC**
Features as Z88 but with 128k RAM Pack, Mains adaptor, Carry case, PC Link. £237

**Z88 Link Pack Mac**
Features as Z88 but with 128k RAM Pack, Mains adaptor. Carry case. Mac Link. £275

**OPTIONS:**-Ram packs, Eprom packs, Mains adaptor. BBC link, PC link, MAC link, Cables, Carry Case and Software.

(PCW 10/90)

Dept PCW, Technology Centre,
19 Progress Way, Croydon, Surrey CR0 4XD
CREDIT CARD MAIL ORDER
081-681 7777
FAX: 081-666 0510

ALL PRICES EXCLUDE VAT

JUST PORTABLES